

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO**

Edson Sorato

**CLASSIFICAÇÃO AUTOMÁTICA DE MODULAÇÕES DIGITAIS
USANDO HISTOGRAMAS E MÁQUINAS DE VETORES DE
SUPORTE**

Florianópolis (SC)

2014

Edson Sorato

**CLASSIFICAÇÃO AUTOMÁTICA DE MODULAÇÕES DIGITAIS
USANDO HISTOGRAMAS E MÁQUINAS DE VETORES DE
SUPORTE**

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação para a obtenção do Grau de Mestre em Ciência da Computação.

Orientador: Prof. Dr. José Luís Almada Güntzel

Florianópolis (SC)

2014

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Sorato, Edson

Classificação Automática de Modulações Digitais Usando
Histogramas e Máquinas de Vetores de Suporte / Edson
Sorato ; orientador, José Luís Almada Güntzel -
Florianópolis, SC, 2014.

119 p.

Dissertação (mestrado) - Universidade Federal de Santa
Catarina, Centro Tecnológico. Programa de Pós-Graduação em
Ciência da Computação.

Inclui referências

1. Ciência da Computação. 2. Classificação Automática de
Modulação. 3. Rádio Cognitivo. 4. Máquinas de Vetores de
Suporte. 5. Arquiteturas Dedicadas de Hardware. I. Güntzel,
José Luís Almada. II. Universidade Federal de Santa
Catarina. Programa de Pós-Graduação em Ciência da Computação.
III. Título.

Edson Sorato

**CLASSIFICAÇÃO AUTOMÁTICA DE MODULAÇÕES DIGITAIS
USANDO HISTOGRAMAS E MÁQUINAS DE VETORES DE
SUPORTE**

Esta Dissertação foi julgada aprovada para a obtenção do Título de “Mestre em Ciência da Computação”, e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Florianópolis (SC), 29 de agosto 2014.

Prof. Dr. Ronaldo dos Santos Mello
Coordenador do Programa

Banca Examinadora:

Prof. Dr. José Luís Almada Güntzel
Orientador
Universidade Federal de Santa Catarina

Prof. Dr. Adalbery Rodrigues Castro
Universidade Federal do Pará

Prof. Dr. Luiz Cláudio Villar dos Santos
Universidade Federal de Santa Catarina

Prof. Dr. Djones Vinicius Lettnin
Universidade Federal de Santa Catarina

Aos meus pais, à minha esposa e à minha filha.

AGRADECIMENTOS

Agradeço primeiramente a meus pais, Olga e Ivanor, que sempre souberam me mostrar o caminho mais correto a seguir, muitas vezes permitindo que eu mesmo tomasse a decisão de qual caminho trilhar, mesmo sabendo que iria tropeçar e assim aprender uma lição. Agradeço também à minha esposa, Edna, e à minha filha, Danielly, pelo amor e apoio incondicionais durante as diversas horas que precisei dedicar ao trabalho de mestrado.

Agradeço ao meu orientador, prof. José Luís Güntzel pela excelente orientação, por ter depositado sua confiança em mim e por ter dedicado seu tempo de lazer para fazer revisões em finais de semanas e feriados. Principalmente pelo exemplo de perseverança, organização e dedicação.

Agradeço ao Renan Alves Fonseca, pelas discussões sobre máquinas de aprendizagem e a vital colaboração neste trabalho. Agradeço também aos demais membros do Laboratório de Computação Embarcada (ECL), pelas conversas, ajudas e pelo ambiente de trabalho colaborativo. Dentre estes agradeço em especial àqueles que tiveram envolvimento direto no trabalho de implementação, Renan de Oliveira Netto, Pedro Veit Michael, Paulo Roberto Ferraz M. M. Barbosa e a Eduardo Pickler Fronza, sem o qual não seria possível a conclusão deste trabalho de mestrado.

Agradeço aos membros do Laboratório de Processamento de Sinais (LAPS) da universidade do Pará pelo repasse da infraestrutura necessária para implementação das simulações realizadas. Em especial, gostaria de agradecer aos professores Adelbaro Klautau e Adalbery Rodrigues Castro e ao Claudomir Cardoso pelas conversas e emails trocados.

Agradeço aos membros da banca pelo tempo dedicado para a revisão e pelas observações, as quais enriqueceram este trabalho. suporte do CNPq (via Projeto INCT-Namitec) e CAPES (apoio para apresentação de trabalho (LASCAS)). Finalmente, agradeço ao CNPq (via Projeto INCT-Namitec) e CAPES pelo suporte para apresentação de trabalho (LASCAS2013).

Obrigado.

*A mente que se abre a uma nova ideia jamais voltará
ao seu tamanho original.*

Albert Einstein

RESUMO

A rápida expansão de comunicações sem fio tornaram o espectro de rádio frequência um recurso bastante escasso. O rádio cognitivo surgiu como uma solução viável para contornar essa dificuldade. No contexto de rádio cognitivo, o monitoramento de espectro (*Spectrum Sensing* (SS)) desempenha um papel fundamental. Em particular, a Classificação Automática de Modulações (*Automatic Modulation Classification* (AMC)) é um dos blocos computacionais que mais demanda recursos dentro do SS. Atualmente, as pesquisas em AMC concentram-se no desenvolvimento de métodos rápidos e precisos que sejam capazes de atingir o *throughput* exigido por aplicações contemporâneas. Alguns trabalhos recentes tem proposto o uso de histogramas do sinal de entrada como parâmetros de classificação e Máquinas de Vetores de Suporte (*Support Vector Machine* (SVM)) como técnica de classificação. O presente trabalho apresenta uma avaliação de tal abordagem e busca projetar arquiteturas de hardware dedicadas eficientes para AMC. Na primeira parte deste trabalho, soluções de hardware para classificação de múltiplas classes usando SVM são projetadas e avaliadas. O esquema "um contra todos melhorado" proposto apresentou uma taxa de acertos até 162% maior do que o esquema "um contra um" comumente utilizado, sendo 30,2% mais eficiente em termos de energia consumida para classificação. Na segunda parte do trabalho, o uso de histogramas em conjunto com SVM é avaliado como uma técnica de classificação cega. Isso é realizado através de experimentos que assumem que o receptor não conhece a taxa de dados do transmissor, em oposição aos trabalhos correlatos. Como outra novidade, histogramas bi-dimensionais são propostos e a taxa de acertos resultante é comparada com aquela obtida para histogramas uni-dimensionais. Os experimentos mostraram que os histogramas bi-dimensionais fornecem uma taxa de acertos ligeiramente maior a um custo de hardware equivalente. Para a classificação baseada em histogramas bi-dimensionais, os seguintes parâmetros foram avaliados: tamanho do conjunto de treino, robustez ao ruído e relação entre número de amostras e taxa de acertos. Para as condições assumidas nos experimentos conduzidos, um tamanho do conjunto de treino de 256 amostras forneceram a melhor relação entre a taxa de acertos e recursos de hardware. Em relação à robustez ao ruído, os experimentos mostraram que a taxa de acertos global é melhorada quando as SVMs são treinadas para toda a faixa de relação sinal-ruído na qual atuarão, ao invés de treinadas para um valor específico de relação sinal-ruído. Finalmente, percebeu-se que a taxa de acertos aumenta drasticamente quando o número de amostras para classificação é maior.

Palavras-chave: Classificação Automática de Modulação (AMC), Rádio De-

finido por Software (SDR), Rádio Cognitivo (CR), Máquina de Vetores de Suporte (SVM), histogramas, arquiteturas de hardware dedicadas, FPGA.

ABSTRACT

The fast expansion of wireless communications made the radio frequency spectrum a quite scarce resource. Cognitive radio emerges as a viable solution to circumvent this difficulty. In the context of cognitive radio, Spectrum Sensing (SS) plays a fundamental role. In particular, Automatic Modulation Classification (AMC) is one of the most computational demanding blocks within SS. Current research in AMC focuses on developing fast and accurate methods that are able to achieve the throughput required by contemporary applications. Some recent works have proposed the use of histograms of the input signal as classification parameters, and Support Vector Machine (SVM) as a classification technique. This work presents an evaluation of such approach looking for designing efficient dedicated hardware architectures for AMC. In the first part of the work, hardware solutions for classifying multiple classes using SVM are designed and evaluated. The proposed "enhanced one against all" scheme presented up to 162% higher accuracy than the commonly used "one against one" scheme, while consuming 30.2% less energy per classification. In the second part of the work, the use of histograms along with SVM is evaluated as a blind classification technique. This was accomplished by conducting experiments assuming that the receiver does not know the transmitted baud rate, in opposition to the correlated work. As another novelty, two-dimensional histograms are proposed and the resulting accuracy is compared to that obtained from one-dimensional histograms. The experiments showed that two-dimensional histograms provide slightly higher accuracy at equivalent hardware costs. The following parameters were evaluated for two-dimensional histogram-based classification: training set size, robustness to noise and relation between number of samples and hit rate. For the conditions assumed in the conducted experiments, a training set size of 256 samples provided the best trade-off between accuracy and hardware resources. Concerning the robustness to noise, the experiments showed that the global accuracy is improved when SVMs are trained for the whole signal-to-noise range that they are supposed to operate, instead of training to a specific signal-to-noise value. Finally, it has been noticed that accuracy is drastically improved when the number of samples for classification is increased.

Keywords: Automatic Modulation Classification (AMC), Software-Defined Radio (SDR), Cognitive Radio (CR), Support Vector Machine (SVM), histograms, dedicated hardware architectures, FPGA.

LISTA DE FIGURAS

Figura 1	Modelo de arquitetura SDR (RIYANTO et al.,)	30
Figura 2	Diagrama em blocos de um classificador de modulações baseado em características	36
Figura 3	plataforma de hardware utilizado pelos <i>frameworks</i> apresentado por Roy et al. (2014) ¹ .	38
Figura 4	Fluxo de dados assumido para receptor SDR.	39
Figura 5	Exemplo de digitalização de um sinal de rádio usando heteródino.	44
Figura 6	Diagrama de constelação de modulações PAM4, PSK4, QAM4, PSK8, QAM16 e QAM64 em condições ideais.	47
Figura 7	Diagramas de constelação da modulação PSK8 com diferentes taxas de amostragem. A SNR é ajustada para infinito e um <i>offset</i> de fase de $\pi/2$ é aplicado.	48
Figura 8	Diagrama em blocos de AMC (DOBRE et al., 2007)	49
Figura 9	Conjunto de dados linearmente separável (BURGES, 1998)...	53
Figura 10	Conjunto de dados não separáveis linearmente (BURGES, 1998)	55
Figura 11	<i>Datapath</i> de um bloco SVM.	73
Figura 12	Diagrama em blocos do classificador baseado no esquema <i>one against one</i> .	75
Figura 13	Hardware do bloco Votador .	76
Figura 14	Diagrama em blocos do esquema <i>one against All</i> .	77
Figura 15	Taxa de acertos dos classificadores baseados em <i>One Against One (OAO)</i> e <i>One Against All (OAA)</i> .	80
Figura 16	Diagrama em blocos do Verificador melhorado .	82
Figura 17	Taxa de acertos dos classificadores baseados em OAO, OAA e OAA melhorado.	82
Figura 18	Histogramas 2-D de sinais com modulações PSK4, PAM4, QAM4, PSK8, QAM16, QAM64 e SNR = 10 dB.	88
Figura 19	Histogramas 2-D de sinais com modulações PSK4, PAM4, QAM4, PSK8, QAM16, QAM64 e SNR = -10 dB.	88
Figura 20	Construção de histogramas 2-D usando informações 1-D com SNR = 10 dB.	89
Figura 21	Construção de histogramas 2-D usando informações 1-D com SNR = -10dB.	89

Figura 22 Diagrama em blocos dos <i>scripts</i> Matlab.	90
Figura 23 Matriz representando um exemplo de conjunto de treino com histogramas 2-D. Cada histograma ocupa exatamente uma linha da ordenada (<i>training sample</i>), totalizando 100 histogramas para cada uma das seis modulações consideradas: PSK4 (linhas 1 a 100), PAM4 (linhas 101 a 200), QAM4 (linhas 201 a 300), PSK8 (linhas 301 a 400), QAM16 (linhas 401 a 500) e QAM64 (linhas 501 a 600). Em cada histograma, os pontos mais escuros significam altos valores de contagem.	93
Figura 24 Matriz representando um exemplo de conjunto de treino com histogramas 1-D. Cada histograma ocupa exatamente uma linha da ordenada (<i>training sample</i>), totalizando 100 histogramas para cada uma das seis modulações consideradas: PSK4 (linhas 1 a 100), PAM4 (linhas 101 a 200), QAM4 (linhas 201 a 300), PSK8 (linhas 301 a 400), QAM16 (linhas 401 a 500) e QAM64 (linhas 501 a 600). Em cada histograma, os pontos mais escuros significam altos valores de contagem.	94
Figura 25 Conjunto parcial do treino da modulação QAM4.	95
Figura 26 Taxa de acertos da classificação para diferentes tamanhos do conjunto de treino.	97
Figura 27 Taxa de acertos da classificação para <i>frameworks multi-condition</i> e <i>matched</i>	99
Figura 28 Taxa de acertos da SVM para diferentes <i>comprimentos da sequência</i> em função da <i>Signal-to-Noise Ratio (SNR)</i>	101
Figura 29 Taxa de acertos da SVM para diferentes valores de <i>nb_bins</i> . .	102
Figura 30 Taxa de acertos para histogramas 1-D para diferentes valores de <i>nb_bins</i>	104
Figura 31 Diagrama em blocos nível RTL de um bloco SVM.	105
Figura 32 Diagrama de blocos de um classificador SVM-HISTO-2D para as 6 modulações digitais consideradas neste capítulo.	106

LISTA DE TABELAS

Tabela 1	Tabela comparativa dos trabalhos apresentados	67
Tabela 2	Resultados da síntese lógica OAO X OAA.	78
Tabela 3	Comparação da síntese lógica OAO X OAA, 16 X 32 <i>bins</i> para módulo.	79
Tabela 4	Comparação de resultados da síntese lógica para as arquiteturas baseadas em OAO, OAA e OAA <i>melhorado</i>	83
Tabela 5	Taxa de acertos global para histogramas 1D e histogramas 2D	103
Tabela 6	Utilização de bits de memória do <i>Field-Programmable Gate Array</i> (FPGA).	106
Tabela 7	Utilização de <i>logical elements</i> nos classificadores SVM-HISTO-1D e SVM-HISTO-2D	107
Tabela 8	Utilização de bits de memória para <i>Histogramas 1-D</i> (HISTO-1D) e <i>Histogramas 2-D</i> (HISTO-2D).	108

LISTA DE ABREVIATURAS E SIGLAS

3G <i>Third Generation</i>	29
3GPP LTE <i>3rd Generation Partnership Project Long-term Evolution</i> ...	28
ALRT <i>Average Likelihood Ratio Test</i>	59
AM <i>Amplitude Modulation</i>	61
AMC <i>Automatic Modulation Classification</i>	111
ASIC <i>Application-Specific Integrated Circuit</i>	28
ASK <i>Amplitude-shift Keying</i>	51
AWGN <i>Additive White Gaussian Noise</i>	60
BPSK <i>Binary Phase-shift Keying</i>	51
CAF <i>Cyclic Autocorrelation Function</i>	63
CDMA-2000 <i>Code Division Multiple Access 2000</i>	29
CR <i>Cognitive Radio</i>	111
CSS <i>Concatenated Sorted Symbols</i>	64
DAGSVM <i>Directed Acyclic Graph SVM</i>	57
DDC <i>Digital Down-converter</i>	44
DFT <i>Discrete Fourier Transform</i>	51
DLRT <i>Discrete Likelihood Ratio Test</i>	60
DS <i>Downsampling</i>	39
DSB <i>Double Sideband</i>	61
DSP <i>Digital Signal Processor</i>	44
DVB-H <i>Digital Video Broadcasting - Handheld</i>	29
DVB-T <i>Digital Video Broadcasting - Terrestrial</i>	29
FCC <i>Federal Communications Commission</i>	27
FM <i>Frequency Modulation</i>	61
FPGA <i>Field-Programmable Gate Array</i>	85
FSK <i>Frequency-shift Keying</i>	51
GLRT <i>Generalized Likelihood Ratio Test</i>	50
GSM <i>Global System for Mobile Communications</i>	28
HISTO-1D <i>Histograms 1-D</i>	85

HISTO-2D <i>Histogramas 2-D</i>	85
HLRT <i>Hybrid Likelihood Ratio Test</i>	50
HSDPA <i>High-speed Downlink Packet Access</i>	28
HSUPA <i>High-speed Uplink Packet Access</i>	29
IF <i>Intermediate Frequency</i>	43
IQ <i>In-phase and In-quadrature</i>	39
ITU <i>International Telecommunication Union</i>	27
LAN <i>Local Area Network</i>	29
LF <i>Likelihood Function</i>	50
LNA <i>Low Noise Amplifier</i>	44
LO <i>Local Oscillator</i>	44
LPF <i>Low Pass Filter</i>	39
LSB <i>Lower Sideband</i>	61
LUT <i>Look-up Table</i>	60
M-PSK <i>M-ary Phase-shift Keying</i>	51
NCO <i>Numerically-controlled Oscillator</i>	44
OAA <i>One Against All</i>	111
OAQ <i>One Against One</i>	69
PAM <i>Pulse-amplitude Modulation</i>	51
PDF <i>Probability Density Function</i>	59
PSK <i>Phase-shift Keying</i>	51
QAM <i>Quadrature Amplitude Modulation</i>	51
RBF <i>Radial Basis Function</i>	62
RF <i>Rádio Frequências</i>	43
SAW <i>Surface Acoustic Wave</i>	44
SCF <i>Spectral Correlation Function</i>	63
SDR <i>Software-Defined Radio</i>	112
SNR <i>Signal-to-Noise Ratio</i>	112
SVM <i>Support Vector Machine</i>	111
SS <i>Spectrum Sensing</i>	111
USB <i>Upper Sideband</i>	61
USRP <i>Universal Software Radio Peripheral</i>	114

WEKA	<i>Waikato Environment for Knowledge Analysis</i>	91
WCDMA	<i>Wideband Code Division Multiple Access</i>	28
WiMax	<i>Worldwide Interoperability for Microwave Access</i>	29

SUMÁRIO

1	INTRODUÇÃO	27
1.1	MOTIVAÇÃO	27
1.2	RÁDIO DEFINIDO POR SOFTWARE - <i>SOFTWARE DEFINED RADIO (SDR)</i>	28
1.3	RÁDIO COGNITIVO - <i>COGNITIVE RADIO (CR)</i>	31
1.3.1	Etapas do Rádio Cognitivo - CR	31
1.4	MONITORAMENTO DO ESPECTRO - SPECTRUM SENSING (SS)	32
1.4.1	Classificação Automática de Modulações - Automatic Modulation Classification (AMC)	35
1.5	ARQUITETURAS PARA SDR E CR	37
1.6	OBJETIVOS DESTE TRABALHO	40
1.7	CONTRIBUIÇÕES	40
1.8	ORGANIZAÇÃO DA DISSERTAÇÃO	41
2	CONCEITOS BÁSICOS SOBRE CLASSIFICAÇÃO AUTOMÁTICA DE MODULAÇÃO (AMC) E SUPPORT VECTOR MACHINE (SVM)	43
2.1	MODELAGEM DO SINAL	43
2.2	CLASSIFICAÇÃO AUTOMÁTICA DE MODULAÇÕES (AMC)	48
2.3	INTRODUÇÃO ÀS MÁQUINAS DE VETOR DE SUPORTE - SUPPORT VECTOR MACHINES (SVM)	52
2.3.1	Caso Linearmente Separável	52
2.3.2	Caso Não Separável Linearmente	54
2.3.3	Combinando Extrator de Parâmetros Baseado em Histograma e SVM Binária Linear para Realizar AMC	55
2.3.4	O Problema de Classificação de Múltiplas Classes usando SVM	56
3	REVISÃO DE TRABALHOS CORRELATOS	59
3.1	CLASSIFICADOR BASEADO NA FUNÇÃO DO <i>TESTE DA TAXA MÉDIA DE VEROSSIMILHANÇA (ALRT)</i>	59
3.2	CLASSIFICADORES BASEADOS EM MÁQUINAS DE VETORES DE SUPORTE (SUPPORT VECTOR MACHINES - SVM)	61
3.2.1	Xiao et al. (2012)	61
3.2.2	Tabatabaei, Krishnan e Anpalagan (2010)	62
3.2.3	Wu et al. (2005)	63
3.2.4	Hu, Song e Wang (2008)	63

3.2.5	Muller, Cardoso e Klautau (2011)	64
3.2.6	Cardoso, Castro e Klautau (2013)	65
3.3	CONCLUSÕES DO CAPÍTULO	66
4	CLASSIFICADORES BASEADOS EM SVM BINÁRIA LINEAR E HISTOGRAMAS 1-D	69
4.1	ANÁLISES NUMÉRICAS PARA DEFINIÇÃO DE ARQUITETURAS DEDICADAS PARA CLASSIFICADORES SVM-HISTO 1-D	70
4.2	<i>DATAPATH</i> DE UM BLOCO <i>SVM LINEAR</i> USANDO HISTOGRAMA 1-D	72
4.3	ARQUITETURAS SVM-HISTO COM ESQUEMA DE DECISÃO <i>ONE AGAINST ONE</i> (OAO)	74
4.4	ARQUITETURAS SVM-HISTO COM ESQUEMA <i>ONE AGAINST ALL</i> (OAA)	76
4.5	COMPARAÇÃO DOS RESULTADOS DA SÍNTESE DE ARQUITETURAS SVM-HISTO UTILIZANDO ESQUEMA OAO COM ARQUITETURAS SVM-HISTO UTILIZANDO ESQUEMA OAA	78
4.6	OTIMIZAÇÃO DA ARQUITETURA BASEADA NO ESQUEMA OAA	81
4.7	CONCLUSÕES DO CAPÍTULO	84
5	USO DE CLASSIFICADORES BASEADOS EM SVM E HISTOGRAMAS 2-D EM UM AMBIENTE NÃO COOPERATIVO	85
5.1	ANÁLISE DA MODELAGEM DE SINAIS USANDO HISTOGRAMAS 1-D E 2-D	86
5.2	INFRAESTRUTURA PARA ANÁLISE NUMÉRICA DE CLASSIFICADORES BASEADOS EM SVM E HISTOGRAMAS 2-D	90
5.3	AVALIAÇÃO DA TAXA DE ACERTOS E DA ROBUSTEZ DE AMC BASEADA EM HISTOGRAMAS 2-D	91
5.3.1	Tamanho do Conjunto de Treino	95
5.3.2	Robustez ao Ruído	98
5.3.3	Tempo e Taxa de Acertos da Classificação	99
5.3.4	Granularidade do Histograma	101
5.4	COMPARAÇÃO DA INFLUÊNCIA DA GRANULARIDADE DE HISTOGRAMAS 1-D VERSUS HISTOGRAMAS 2-D NA TAXA DE ACERTOS DA CLASSIFICAÇÃO	102
5.5	ARQUITETURAS PARA CLASSIFICADORES SVM E HISTOGRAMAS 2-D	104

5.5.1	Comparação dos Resultados de Síntese para HISTO-1D e HISTO-2D	107
5.6	CONCLUSÕES DO CAPÍTULO	108
6	CONCLUSÕES	111
6.1	TRABALHOS FUTUROS	113
	REFERÊNCIAS	115

1 INTRODUÇÃO

Este capítulo visa apresentar o problema de classificação de modulação digital, objeto desta dissertação, destacando sua relevância para projetos de sistemas de rádio cognitivo contemporâneos. Inicialmente, são apresentadas a motivação e uma breve contextualização do tema proposto. Em seguida, a estrutura genérica de um sistema de rádio cognitivo contemporâneo é apresentada, indicando a etapa em que o tema desta dissertação se encaixa. Na sequência, o problema de classificação de modulação digital é formalmente apresentado. Finalmente, são apresentados: a justificativa do tema, o escopo da dissertação, a plataforma para rádio definido por software, as principais contribuições científicas e a organização do texto.

1.1 MOTIVAÇÃO

Uma das recentes realizações da humanidade foi a invenção de novos e mais eficientes meios de comunicação que vencem facilmente a barreira do espaço. Dentre esses novos meios de comunicação, a comunicação sem fio foi uma das que mais evoluíram nos últimos anos. Porém, o aumento da demanda por novos meios de comunicação levou a uma intensa utilização do espectro de *Rádio Frequências* (RF). Devido a esse fato, tornou-se importante a criação de regras de utilização e de fiscalização deste recurso, regras essas que são definidas e impostas por dispositivos reguladores subordinados ao governo de cada país, seguindo acordos estabelecidos no *International Telecommunication Union* (ITU) ¹. O principal intuito desses dispositivos reguladores é garantir que os chamados usuários primários, que têm o direito de concessão de uso de uma determinada faixa do espectro de RF, não sejam prejudicados por transmissões não autorizadas que porventura usem a mesma faixa. Estas últimas são classificadas como transmissões de usuários secundários.

Esses dois fatores em conjunto, a intensa utilização do espectro de RF e a regulamentação imposta pelos governos, tornaram o espectro de RF um recurso escasso e disputado. Entretanto, um estudo realizado pelo órgão governamental norte americano *Federal Communications Commission* (FCC) na cidade de New York no ano de 2002 demonstrou que o espectro de frequências está subutilizado quando se consideram aspectos temporais, geográficos e as faixas de frequência utilizadas. Este mesmo estudo demonstrou que, em

¹ Agência das Nações Unidas que tem o propósito de coordenar operações e serviços de telecomunicações em nível mundial.

média, a utilização do espectro varia entre 15% e 80% em condições normais (exceto em estados de emergência, quando a utilização pode chegar a 100%) (KOŁODZY, 2002). A partir desta constatação, tem sido desenvolvidas formas de otimizar a utilização do espectro, de modo a permitir que mais usuários possam ser atendidos com novos serviços.

São inúmeras as linhas de pesquisa na área de telecomunicações que buscam técnicas que permitam um uso mais eficiente do espectro de RF. Uma destas linhas de pesquisa é o *Cognitive Radio* (CR), que tem como principal objetivo possibilitar que um usuário secundário possa utilizar uma faixa do espectro de RF licenciada. Para tanto, sistemas de CR devem garantir que não interferirão nas transmissões dos usuários primários.

O surgimento do conceito do CR se deve à evolução natural do *Software-Defined Radio* (SDR) e do surgimento de antenas reprogramáveis, as quais possibilitam tornar a recepção do sinal mais seletiva.

1.2 RÁDIO DEFINIDO POR SOFTWARE - *SOFTWARE DEFINED RADIO* (SDR)

As soluções de hardware para rádio comunicações sofreram constantes transformações ao longo do tempo. Os primeiros sistemas de comunicação sem fio digitais trafegavam dados em um único padrão e isso permitiu a utilização de circuitos integrados de aplicação específica - *Application-Specific Integrated Circuits* (ASICs), energeticamente mais eficientes do que os *Field-Programmable Gate Arrays* (FPGAs) e microprocessadores de propósito geral.

A proliferação de padrões de comunicações sem fio impedem a utilização da abordagem tradicional baseada em ASIC para cada padrão devido ao aumento do custo para suportar todos os padrões. Para solucionar esse problema os fabricantes de sistemas de comunicação por rádio foram forçados a desenvolver soluções que suportam múltiplos padrões através da reutilização do hardware de forma a manter o consumo de energia em um nível aceitável para uso em dispositivos móveis (RAMACHER et al., 2007).

O surgimento do SDR se deve à evolução dos sistemas de comunicação sem fio e à consequente proliferação de padrões de comunicação que permitem trafegar mais dados ocupando uma banda menor do espectro de frequências. Padrões como *Global System for Mobile Communications* (GSM), *Wideband Code Division Multiple Access* (WCDMA), *High-speed Downlink Packet Access* (HSDPA) e *3rd Generation Partnership Project Long-term Evolution* (3GPP LTE) são suportados nos atuais sistemas de telefonia celular ou estão previstos para serem suportados no futuro. O sis-

tema *Third Generation* (3G), por exemplo, já prevê padrões como WCDMA/UMTS (*Universal Mobile Telecommunications System*), HSDPA, *High-speed Uplink Packet Access* (HSUPA) e *Code Division Multiple Access 2000* (CDMA-2000) (ANJUM et al., 2011).

Além disso, esses dispositivos devem possuir capacidade de gerenciamento e comunicação em outras redes como *Worldwide Interoperability for Microwave Access* (WiMax), *Local Area Network* (LAN), *Digital Video Broadcasting - Terrestrial* (DVB-T) e *Digital Video Broadcasting - Handheld* (DVB-H) (ANJUM et al., 2011)(DELAHAYE et al., 2004).

O receptor ideal do SDR proposto por Mitola prevê que a conversão analógica para digital seja feita no nível de sinal de rádio frequências imediatamente após a antena. Similarmente, no lado do transmissor, o sinal digital é entregue ao conversor digital para analógico na etapa de amplificação para a antena sem novas etapas intermediárias. O conceito ideal não é implementável com a tecnologia atual. Um dos problemas a ser contornado é o da antena a qual, segundo a definição teórica do SDR, deve apresentar características uniformes em uma ampla faixa de frequências (ARSLAN, 2007). Além disso, deve permitir exploração espacial de forma a possibilitar a busca por oportunidades no espectro de frequências. Estes requisitos de projeto impulsionam as pesquisas das chamadas "antenas inteligentes", que permitem reconfigurabilidade a partir de um controle em software (ARSLAN, 2007). Outro impeditivo que também tem demandado grande esforço de pesquisa são os conversores A/D e D/A atuais, os quais não possuem a velocidade e taxa dinâmica necessárias para possibilitar conversões em frequências de RF.

A Figura 1 mostra o diagrama de blocos de um SDR genérico. O conceito básico por trás do SDR é que tarefas normalmente realizadas por um hardware dedicado (como por exemplo filtragem, amplificação, modulação/demodulação, detecção, etc) passem a ser realizadas em software, permitindo desta forma o reuso do hardware. Uma das vantagens desta abordagem é que estes componentes podem ser reprogramados em campo após a venda e instalação do dispositivo, o que permite aos fabricantes destes dispositivos uma grande versatilidade.

No entanto, tais operações não podem ser realizadas diretamente nas frequências nas quais os sinais são transmitidos (banda de transmissão) e portanto, uma fase de conversão para uma frequência mais baixa (banda básica) é necessária. Esta etapa é realizada em hardware analógico e é representada pelo bloco "*Wideband RF front End*" na Figura 1. Uma vez feita a conversão do sinal para banda básica, o sinal pode então ser convertido para o domínio digital. Na Figura 1 o bloco *ADC/DAC* delimita a fronteira do processamento analógico e digital. No lado digital, as etapas necessárias para o processamento dos sinais podem ser realizadas em software (ou em blocos acelerado-

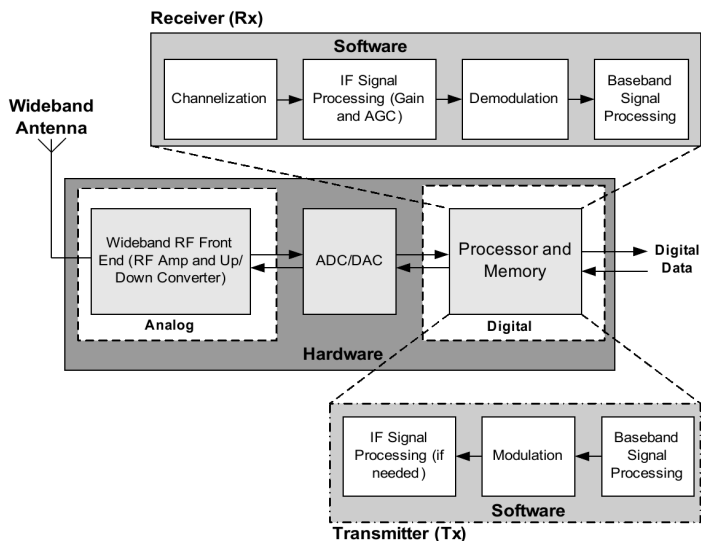


Figura 1 – Modelo de arquitetura SDR (RIYANTO et al.,)

res dedicados). A Figura 1 mostra, de forma idealizada, os processos necessários para a transmissão e recepção do sinal. Na transmissão, o conjunto de bits a ser transmitido passa pelo bloco "*Baseband Signal Processing*", onde é realizado o pré-processamento necessário para a transmissão (inserção de códigos de correção de erro, empacotamento, etc). Na etapa seguinte, o conjunto de bits é transformado em símbolos (de acordo com a constelação da modulação utilizada) pelo bloco "*Modulation*" e finalmente, é transformado em sinal de RF em banda básica pelo bloco "*IF Signal Processing*". Esse sinal é convertido para o domínio analógico pelo bloco "*ADC/DAC*" e transmitido após passar pelo bloco "*Wideband RF front End*", que o converte para a banda de transmissão. No lado do receptor, após passar pelo bloco "*Wideband RF front End*" para ser convertido para banda básica e ser amostrado pelo bloco "*ADC/DAC*", o sinal recebido é filtrado pelo bloco "*Channelization*" de forma a obter somente a componente do sinal dentro da banda (canal) desejada. Logo após, o sinal passa pelo bloco "*IF Signal Processing*" que faz os ajustes de ganho necessários para sua correta recuperação. Já na banda de frequência do canal desejada e com os ganhos ajustados, o sinal passa pelo bloco "*Demodulation*", que extrai o conjunto de bits de informação, juntamente com os bits de correção de erro e demais dados necessários para a correta recuperação do sinal inseridos na transmissão. Esse conjunto de bits

é processado pelo bloco " *Baseband Signal Processing*", que entrega em sua saída o dado recebido recuperado.

1.3 RÁDIO COGNITIVO - *COGNITIVE RADIO (CR)*

Segundo a FCC, a definição de CR é "Um rádio ou sistema de rádio comunicação que tem consciência de seu ambiente eletromagnético operacional e que pode ajustar seus parâmetros de operação automaticamente e autonomamente de forma a maximizar o *throughput*, mitigar interferências, facilitar a interoperabilidade e acessar o mercado secundário" (FCC, 2005), (YUCEK; ARSLAN, 2009).

Para realizar tal tarefa, o CR precisa ter a capacidade de "sentir" o ambiente a sua volta e realizar alterações em suas próprias características de transmissão de forma a não interferir nas transmissões dos usuários primários. Portanto, um CR deve detectar de uma forma rápida e precisa a presença de um usuário primário e, uma vez detectada a presença deste, deve alterar um ou mais dos seguintes parâmetros:

- frequência de transmissão;
- potência de transmissão;
- esquema de modulação utilizado;
- ângulo de transmissão;
- algum outro parâmetro que garanta que o CR não cause interferências no usuário primário.

Um exemplo de CR é o padrão **IEEE 802.22**, atualmente em vigor nos Estados Unidos, que viabiliza a utilização, por usuários secundários, de frequências destinadas a transmissões televisivas (54MHz até 862MHz).

1.3.1 Etapas do Rádio Cognitivo - CR

Um sistema de CR é composto de quatro grandes tarefas ou etapas, quais sejam:

- **Monitoramento do espectro** - processo responsável por procurar faixas de frequências não ocupadas e por detectar a entrada de usuários primários na frequência de operação do CR;

- **Gerenciamento do espectro** - processo responsável por buscar constantemente o melhor canal para transmissão;
- **Mobilidade** - processo responsável por possibilitar que um usuário de CR se desloque geograficamente através do estabelecimento de novos *links* antes que o *link* atual seja desfeito;
- **Compartilhamento do espectro** - processo responsável por viabilizar a interoperabilidade de vários sistemas de CR em faixas de frequências adjacentes, evitando assim a colisão entre diferentes CR.

Dentre estas tarefas, o **monitoramento do espectro** é de especial interesse para esta dissertação e por isto será apresentada em detalhes na seção que segue.

1.4 MONITORAMENTO DO ESPECTRO - SPECTRUM SENSING (SS)

O monitoramento do espectro é a tarefa que demanda maior esforço computacional. Ela permite ao CR obter "consciência" do espectro eletromagnético a sua volta e identificar a presença de usuários primários. Essa "consciência" pode ser obtida utilizando geolocalização e uma base de dados, pacotes especiais para gerenciamento da rede (*beacons*), ou monitoramento local nos CR (YUCEK; ARSLAN, 2009).

Apesar de o termo "monitoramento de espectro" ser tradicionalmente entendido como medidas de conteúdo espectral ou de energia de rádio frequências sobre o espectro, quando se considera CR, tal termo tem uma abrangência maior que envolve obter características de uso do espectro em múltiplas dimensões como tempo, espaço, frequência e código. Também envolve a determinação de quais tipos de sinais estão ocupando o espectro, incluindo a modulação, forma de onda, largura de banda, frequência da portadora etc. Isto requer uma técnica de análise de sinal mais poderosa, com uma alta complexidade computacional (YUCEK; ARSLAN, 2009).

No nível de hardware, são muitos os requisitos para realizar o monitoramento de espectro. Dentre estes, citam-se (YUCEK; ARSLAN, 2009):

- Taxa de amostragem alta;
- A/D com alta resolução e larga faixa dinâmica;
- Processadores de sinal de alto *throughput*.

Um outro desafio enfrentado pelo CR é o fato de que um usuário primário pode começar a transmitir a qualquer momento e com uma certa duração desconhecida pelo CR. Nesta situação, o CR deverá identificar o usuário

primário dentro do intervalo de tempo em que este estiver transmitindo. Desta forma, a seleção de parâmetros para o monitoramento do espectro é um compromisso entre a velocidade (tempo de monitoramento) e a confiabilidade dos resultados deste monitoramento. Parâmetros como frequência de monitoramento (quantas vezes o CR faz uma determinada investigação), tempo para a detecção de um usuário primário em determinado canal, tempo para sair do canal e outros parâmetros devem ser escolhidos com cuidado (YUCEK; ARSLAN, 2009). Outro fator a ser levado em conta é a tolerância à interferências por parte do usuário primário.

Segundo Yucek e Arslan (2009), os métodos mais comuns de monitoramento do espectro são baseados em:

- Energia do sinal;
- Forma de onda;
- Cicloestacionaridade;
- Identificação de rádio;
- Filtros casados.

Monitoramento com detector baseado em energia do sinal, também chamado de radiometria ou periodograma, é mais comum de ser encontrado devido ao baixo custo computacional e baixa complexidade de implementação. É considerado um método mais genérico do que os demais métodos. O sinal é detectado através da comparação da saída de um detector de energia com um limiar. Alguns problemas conhecidos deste tipo de detector são:

- Determinação do limiar para detecção. O limiar de detecção é fortemente dependente do ruído mínimo (*floor*) do sistema;
- Incapacidade para diferenciar interferência de usuário primário e ruído;
- Baixo desempenho em casos de *Signal-to-Noise Ratio* (SNR) baixas.

No **monitoramento baseado em formas de onda**, padrões do sinal recebido, como por exemplo o preâmbulo, midâmbulo, padrões de sinal piloto e outros, são correlacionados com cópias conhecidas desses próprios padrões de sinal de forma a obter seu reconhecimento. Esse método, no entanto, é capaz de identificar apenas o universo de sinais para os quais possui informação prévia que possam caracterizá-los. Por esse motivo, é chamado de monitoramento coerente de sinais. Esse método é considerado mais eficiente, em termos de confiabilidade e tempo de convergência, do que o método baseado

em detector de energia. Outra característica deste método é que o desempenho do algoritmo de monitoramento aumenta à medida que o comprimento (tempo de duração) do padrão do sinal aumenta.

Monitoramento baseado em cicloestacionaridade utiliza características cicloestacionárias do sinal recebido. Características cicloestacionárias são causadas por periodicidade no sinal ou, indiretamente, em alguns de seus elementos estatísticos. Por exemplo, sinais modulados podem apresentar componentes não randômicos como unilateralidade dupla (devido a portadora senoidal) e taxa de chaveamento (devido ao período do símbolo). Esse método de monitoramento pode diferenciar usuários primários de ruído do sinal devido ao fato do ruído ser estacionário no sentido amplo (*Wide Sense Stationary* - WSS), sem autocorrelação ², enquanto que sinais modulados são cicloestacionários, com autocorrelação espectral devido à redundância dos padrões periódicos do sinal. Outra vantagem é que características cicloestacionárias podem ser utilizadas para distinguir entre diferentes tipos de transmissões de usuários primários.

O problema apresentado pelo monitoramento baseado em cicloestacionaridade, além do alto custo computacional para extração de parâmetros (geralmente baseado em funções estatísticas), é o fato do método falhar completamente na presença do efeito de desvanecimento do canal.

Monitoramento baseado em identificação de rádio consiste em identificar a tecnologia de rádio utilizada pelo usuário primário. Essa técnica permite uma maior exploração dimensional (espaço, tempo, frequência, ângulo de chegada, etc) e uma maior precisão na identificação do sinal recebido. Por exemplo, ao identificar que um sinal é do tipo *bluetooth*, o CR pode inferir que o usuário primário está em um raio de 10 metros (alcance típico para o *bluetooth*). No monitoramento baseado em identificação de rádio, diversas características podem ser extraídas do sinal recebido e usadas para selecionar a tecnologia empregada pelo usuário primário mais provável. Diferentes métodos de classificação podem ser utilizados como método de seleção, como por exemplo, redes neurais, classificadores bayesianos³ etc. Exemplos de características são: a quantidade de energia detectada e sua distribuição no espectro, a largura do canal e sua forma, faixa de operação e a frequência central do sinal recebido, etc.

²Autocorrelação refere-se ao grau de "coincidência" de um determinado sinal com uma versão atrasada do mesmo sinal. A função de autocorrelação fornece uma medida de quão "aproximado" é um sinal de uma versão do próprio sinal deslocado de τ unidades de tempo (SKLAR, 2001).

³Classificadores bayesianos são classificadores estatísticos que utilizam a probabilidade conhecida (probabilidade *a priori*) associada a uma classe (determinada através de treinamento) para determinar a probabilidade desconhecida (probabilidade *a posteriori*) de um novo elemento (desconhecido) pertencer àquela classe (HANSON; STUTZ; CHEESEMAN, 1991).

Monitoramento baseado em filtros casados é um método que é considerado ótimo se o sinal do usuário primário é totalmente conhecido. A principal vantagem é o curto tempo para identificação de falso alarme, ou erro de classificação, quando comparado com os demais métodos apresentados. A principal desvantagem deste método reside no fato de ser necessário demodular o sinal recebido e isso significa conhecer completamente as características do usuário primário: largura de banda, frequência de operação, tipo de modulação, formato do pulso e formato do *frame*.

No contexto de CR, a implementação de detector em hardware usando monitoramento baseado em filtros casados é inviável devido à exigência de conhecimento completo do sinal do usuário primário. Além disso, tal implementação consumiria muita energia (YUCEK; ARSLAN, 2009). O monitoramento baseado em identificação de rádio exige do CR capacidade de armazenar características específicas das tecnologias identificadas. Este fato pode ser impraticável em situações onde os recursos de hardware são limitados. No caso do monitoramento baseado em cicloestacionaridade, além do alto custo computacional para extração de parâmetros, geralmente baseado em funções estatísticas, existe o problema de o método falhar completamente na presença do efeito de desvanecimento do canal. O monitoramento baseado em detector de energia do sinal não fornece informações sobre o tipo de sinal detectado e portanto, não fornece informações que um sistema de CR necessita para um melhor aproveitamento do espectro de frequências (YUCEK; ARSLAN, 2009).

Para um método de monitoramento factível em hardware, a complexidade computacional da arquitetura resultante deve ser levada em consideração pois influencia diretamente o tamanho do hardware e consequentemente, o consumo de energia. Portanto, o monitoramento baseado em formas de onda parece ser o método mais indicado. Para o caso específico de monitoramento de espectro aplicado ao CR, é necessário extrair uma informação mais completa do método de monitoramento. Essa tarefa é realizada por algoritmos de *Automatic Modulation Classification* (AMC), apresentados na próxima seção.

1.4.1 Classificação Automática de Modulações - Automatic Modulation Classification (AMC)

É importante notar a diferença entre a tarefa de detectar atividade em uma determinada faixa de frequências e a classificação de modulações. No primeiro caso, um detector de energia simples combinado com um esquema para varrer a faixa de frequências de interesse é o suficiente. No caso do classificador de modulações, no entanto, o interesse é obter informações mais

detalhadas do sinal do usuário primário, de forma a permitir uma melhor interação com o meio.

Para identificar o esquema de modulação utilizado pelo usuário primário, um método de classificação de modulação deve ser utilizado. Basicamente, os métodos existentes pertencem a um dentre dois grupos: **baseados em verossimilhança** (*likelihood-based*) e **baseados em parâmetros** (*feature-based*) (DOBRE et al., 2007).

Nos métodos baseados em verossimilhança, a taxa de verossimilhança⁴, calculada através da função de mesmo nome, é comparada com um limiar de decisão. Apesar de este método ser considerado ótimo devido à baixa probabilidade de falsa classificação, apresenta um alto custo computacional (DOBRE et al., 2007), o que torna sua aplicação difícil para implementações em hardware.

Métodos baseados em parâmetros podem utilizar uma variada gama de características extraídas do sinal tais como, a variância do sinal, a transformada *wavelet*, a função de densidade de probabilidade, cumulantes (DOBRE et al., 2007), símbolos ordenados e concatenados e histogramas (CARDOSO; CASTRO; KLAUTAU, 2013). É importante salientar que uma comparação direta entre os métodos é muito difícil por diversas razões. Entre elas está o fato de que cada método utiliza um número de parâmetros diferentes. (DOBRE et al., 2007).

Para os métodos baseados em parâmetros a estrutura básica é a mesma: um bloco de extração de parâmetros (*front-end*) e um bloco classificador (também chamado de *back-end* em algumas literaturas) como mostrado na Figura 2 (CARDOSO; CASTRO; KLAUTAU, 2013). A complexidade da

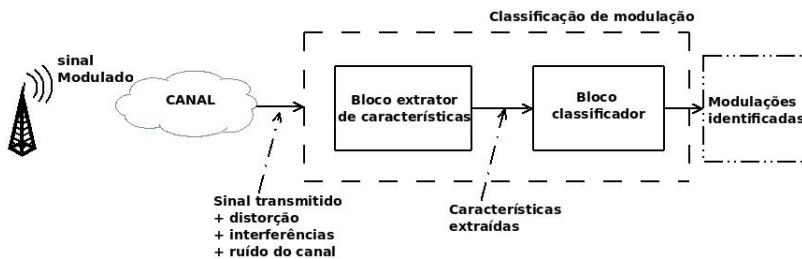


Figura 2 – Diagrama em blocos de um classificador de modulações baseado em características

⁴Em estatística, a estimativa por máxima verossimilhança (*maximum-likelihood estimation-MLE*) é um método para estimar os parâmetros de um modelo estatístico. Assim, a partir de um conjunto de dados e dado um modelo estatístico, a estimativa por máxima verossimilhança estima valores para os diferentes parâmetros do modelo.

etapa de classificação é diretamente influenciada pelo tipo de parâmetro extraído para classificação.

Entre os métodos de classificação existentes, as máquinas de vetor de suporte (*Support Vector Machines* (SVMs)) tem emergido como uma opção atrativa devido ao seu desempenho em muitas tarefas (quando comparado com redes neurais, apresenta melhor capacidade de predição (BYVATOV et al., 2003) com tempo de treino sensivelmente menor (CORTES; VAPNIK, 1995)) e a habilidade de tratar grandes quantidades de dados com um custo computacional aceitável (LUXBURG; SCHÖLKOPF, 2008). Tais características fazem dos classificadores baseados em SVM bons candidatos para a implementação de hardware dedicado para a tarefa de AMC com capacidade de operar em tempo real e por este motivo, foram escolhidos como método de classificação a ser investigado no contexto do presente trabalho.

Para uma boa apreciação das vantagens do uso de SVM para a implementação de classificadores de modulação digital é necessário, antes, entender como o sinal é modelado tanto do lado da transmissão como da recepção. Assim, a próxima seção apresenta a modelagem do sinal, com enfoque em classificação de modulação.

1.5 ARQUITETURAS PARA SDR E CR

As arquiteturas propostas atualmente para CR são as mesmas utilizadas em SDR (ROY et al., 2014). Segundo esses mesmos autores, existem 4 *frameworks* de software mais relevantes utilizados para a implementação de sistemas de CR e SDR:

- GNU Radio;
- OSSIE - *Open Source SCA (Software Communication Architecture) Implementation Embedded*;
- ALOE - *Abstraction Layer and Operating Environment*;
- IRIS (parte do *Irish Research Project*).

Cada um destes *frameworks* apresenta diferentes propostas para viabilizar os serviços necessários para a implementação do CR. Uma característica comum a todos os *frameworks* é a interação entre o processamento realizado em hardware e o processamento realizado em software (sendo que este último geralmente utiliza um processador de propósito geral (GPP)) o que deu origem à camada de abstração de hardware (HAL - *hardware Abstraction Layer*) dentro dos *frameworks* citados. Isso possibilita grande variação na parte de

processamento em hardware (após a digitalização do sinal), conforme pode ser observado na Figura 3.

Platform Design			Framework or Open Environment			
			<i>GNU Radio</i>	<i>OSSIE</i>	<i>ALOE</i>	<i>IRIS</i>
	<i>OS</i>		Linux	Linux	Linux	Linux
	<i>Middleware</i>		no	CORBA	custom	?
	<i>Device</i>	<i>GPP</i>	yes	Yes	yes	Yes
		<i>GPU</i>	Nvidia	Partial	no	Cell ^a
		<i>DSP</i>	Omap	Omap	C6000	Yes
		<i>FPGA</i>	partial	Partial	yes	Yes

a: Cell architecture is a close cousin of GPU's one.

Figura 3 – plataforma de hardware utilizado pelos *frameworks* apresentado por Roy et al. (2014) ⁵.

Além destes *frameworks*, existem iniciativas comerciais que tentam viabilizar o CR, como por exemplo, as soluções propostas pelo *Nokia Research Center* (Nokia, 2014), por XG Technology (2014) e por Ettus Research (2014). Como era de se esperar, o conceito de CR tem maior apelo para dispositivos móveis. No entanto, até o momento, as soluções apresentadas ainda são compostas de conjuntos combinados de processadores de sinais digitais (DSP), FPGAs e processadores de propósito geral (GPP) e, no geral, não são destinadas para uso em dispositivos portáteis. Muitos esforços estão sendo feitos em todos os campos para viabilizar o uso de CRs e SDR em dispositivos portáteis. Um exemplo é o trabalho de Immonen et al. (2012), que apresenta uma proposta para reduzir o número de blocos que lidam com rádio frequência envolvidos na reconfiguração de rádios que suportam múltiplos padrões (*smart phones* contemporâneos). Segundo os autores, a mesma ideia teria aplicação em sistemas de SDR portáteis. Já na etapa digitalizada do processamento do sinal, o trabalho de Ouedraogo, Gautier e Sentiéys (2014) apresenta um método para automatizar a reconfiguração da camada física de um sistema SDR utilizando FPGA. O maior apelo do uso do FPGA para sistemas de CR e SDR está justamente na reconfigurabilidade que o disposi-

⁵CORBA (*Common Object Request Broker Architecture*) - *middleware* que provê um método padronizado para objetos de software se comunicarem independente do processador no qual estão instalados.

tivo permite. Para a maioria destas plataformas, o monitoramento do espectro é feito através de um detector de energia devido, principalmente, ao baixo custo computacional apresentado por esse método. Muitos sistemas propostos, como por exemplo o *GNU Radio*, utilizam como plataforma de hardware, placas de som, presentes em todos os computadores atuais. No entanto, muitas são as iniciativas comerciais, como o *Universal Software Radio Peripheral* (USRP) por exemplo, que utilizam sofisticados esquemas de aquisição e tratamento do sinal geralmente baseados em FPGAs (HARUYAMA, 1999).

A Figura 4 mostra o fluxo de dados assumido no presente trabalho, onde todo o processamento é realizado em uma plataforma de hardware para SDR, com exceção da etapa de processamento em software. Em geral, em uma plataforma de hardware para SDR existe um bloco que realiza os passos intermediários entre o processamento analógico e o processamento em software, e que pode ser realizado por um FPGA.

As três principais funções implementadas no FPGA são (DICK; PEDERSEN, 2001):

- Demodulação Digital em Fase e Quadratura (IQ);
- Filtro passa baixas (LPF);
- *Downsampling* (DS).

Geralmente, *Low Pass Filter* (LPF) e *Downsampling* (DS) são combinados em um bloco único. O passo DS reduz a taxa de dados do sinal, tornando possível a transferência de dados para o bloco de software. Assim, após este passo, são possíveis variações arquiteturais. Diferentes funções podem ser implementadas em hardware ou no lado do software.

Neste trabalho, o passo de AMC é assumido antes do passo de DS, em hardware. De maneira particular, o bloco que realiza AMC é colocado logo após a demodulação *In-phase and In-quadrature* (IQ), como apresentado na Figura 4.

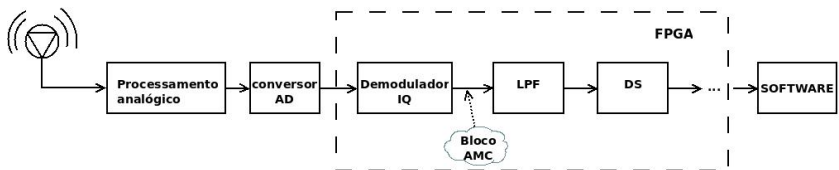


Figura 4 – Fluxo de dados assumido para receptor SDR.

1.6 OBJETIVOS DESTE TRABALHO

O objetivo principal deste trabalho é identificar esquemas de classificação de modulações digitais (AMC) baseados em SVM que possam ser utilizados para implementações eficientes em hardware, a serem utilizadas em um sistema de rádio cognitivo (CR).

Desta forma, os objetivos específicos são:

1. Avaliar a taxa de acertos do esquema de classificação que usa SVMs binárias lineares considerando um conjunto de modulações digitais tipicamente utilizados nos padrões de comunicação contemporâneos e uma faixa de SNR de interesse;
2. Investigar a necessidade de adoção de SVMs com *kernel* não-linear;
3. Projetar, descrever e sintetizar arquiteturas de hardware para o esquema de classificação baseado em SVMs lineares, a fim de se obter estimativas realistas de custo, desempenho e consumo de energia;
4. Investigar o impacto de decisões de projeto das SVMs binárias lineares na taxa de acertos da classificação e no custo, desempenho e energia das respectivas implementações em hardware. Dentre estas decisões encontram-se o esquema de decisão para classificação de múltiplas classes e o número de *bins* a ser utilizado para treino e classificação.

1.7 CONTRIBUIÇÕES

As duas principais contribuições deste trabalho são:

- Avaliação profunda e detalhada da taxa de acertos da classificação usando SVMs binárias lineares (capítulos 4 e 5);
- Avaliação de área, atraso crítico e energia associados às suas implementações em hardware (capítulo 4).

Em especial, os seguintes parâmetros de teste foram consideradas nas avaliações realizadas no capítulo 5, o que diferencia substancialmente o presente trabalho dos trabalhos correlatos anteriores:

1. Foram assumidos 10 fatores de amostragem (8,0; 5,33; 3,47; 2,28; 1,50; 1,0; 0,66; 0,44; 0,29; 0,19)⁶, os quais servem para modelar a

⁶Esta sequência segue uma escala logarítmica, método normalmente utilizado em processamento de sinais para avaliar grandezas relacionadas a frequência (HAYKIN, 2008).

ausência de sincronismo entre o emissor e o receptor, o que corresponde a situações mais realistas. Desta forma, pode-se avaliar também a robustez do método;

2. Ampla faixa de valores de SNR (-40 dB e $+10$ dB);
3. O tamanho do conjunto de características (vetores de treino) usado no treinamento das SVMs foi variado, também para avaliar a robustez do método;
4. Foram testados os seguintes comprimentos de sequências de amostras (número de exemplares nos vetores de treino/teste): 75, 150, 300, 600 e 1200;
5. O número de *bins* utilizados foi variado;
6. Foram testados histogramas unidimensionais e bidimensionais.

Como resultado das avaliações, observou-se que a classificação baseada em SVMs binárias lineares é extremamente robusta e eficiente, uma vez que foram atingidas elevadas taxas de acertos para as modulações escolhidas, mesmo assumindo-se uma grande variação da taxa de amostragem por parte do receptor. Além disso, os resultados das sínteses das arquiteturas de hardware apontam para reduzida área e desempenho compatível com a aplicação-alvo, qual seja, os rádios cognitivos.

1.8 ORGANIZAÇÃO DA DISSERTAÇÃO

Esta dissertação está organizada como segue:

No Capítulo 2 são apresentados conceitos necessários para o entendimento do processo de AMC, iniciando pela modelagem do sinal e evoluindo para métodos de classificação, culminando com classificadores baseados em SVM. Também é apresentado o problema de classificação de múltiplas classes e as abordagens utilizadas para solucioná-lo.

O Capítulo 3 apresenta os trabalhos correlatos, dando destaque para os trabalhos que fazem uso do classificador SVM.

O Capítulo 4 apresenta as avaliações arquiteturais realizadas para classificadores baseados em histograma 1-D.

O Capítulo 5 apresenta a modelagem numérica para avaliações de classificadores baseados em histogramas 2-D. Este capítulo apresenta também uma comparação entre classificadores baseados em histogramas 1-D e histogramas 2-D.

O Capítulo 6 apresenta as conclusões gerais deste trabalho de mestrado e os possíveis trabalhos futuros.

2 CONCEITOS BÁSICOS SOBRE CLASSIFICAÇÃO AUTOMÁTICA DE MODULAÇÃO (AMC) E SUPPORT VECTOR MACHINE (SVM)

Este capítulo apresenta os conceitos básicos sobre classificação automática de modulações (*Automatic Modulation Classification* (AMC)) e sobre máquina de vetor de suporte (*Support Vector Machine* (SVM)). Em especial, abordam-se os aspectos particulares relativos ao uso de SVM binária linear para realizar AMC. Inicialmente, a modelagem do sinal modulado digitalmente é apresentada. Após, os conceitos de AMC são introduzidos e a seguir, a teoria de SVM é apresentada. Finalmente, discute-se o problema da classificação de múltiplas classes com SVM.

2.1 MODELAGEM DO SINAL

Um sinal de rádio $\mathbf{r}(\mathbf{t})$ normalmente tem uma potência baixa, centrado em uma pequena faixa (canal) com uma alta frequência (1,8 GHz por exemplo), necessitando de um pré-processamento antes de ser digitalizado com um conversor Analógico para Digital (A/D). Existem diversas arquiteturas de conversão do sinal em *Rádio Frequências* (RF) para o domínio digital para posteriormente ser processado pelo bloco AMC. Esta seção apresenta a arquitetura adotada neste trabalho, a qual é genérica o suficiente para abranger os principais sistemas de comunicação de interesse de AMC.

Uma grande gama de conversores A/D são capazes de trabalhar com sinais na banda base ou sinais cujo espectro esteja centrado na frequência intermediária (*Intermediate Frequency* (IF)), por exemplo, 70 MHz. É possível ter-se um conversor A/D próximo à antena digitalizando quase diretamente um sinal de RF em 1,8 GHz, por exemplo. Desta forma, o sinal de RF digitalizado pode ser processado diretamente utilizando técnicas de processamento de sinais. Entretanto, com a tecnologia atual, é mais comum o uso de um módulo heteródino¹ e digitalizar o sinal em uma IF ou até mesmo aplicar conversão direta de RF para banda base e então digitalizá-lo. Um esquema baseado em heteródino é apresentado na Figura 5.

Na Figura 5, o sinal é processado da esquerda para a direita: o sinal de RF é recebido pela antena e passado por um filtro passa banda que limita a faixa de frequências à banda de interesse. Em seguida, o sinal passa por um

¹Um receptor heteródino é composto basicamente por um bloco "oscilador local", utilizado para sintonizar o receptor na faixa de frequências de interesse, e um bloco "mixer", que "separa" o sinal desejado dos demais sinais captados pela antena (HAYKIN, 2008).

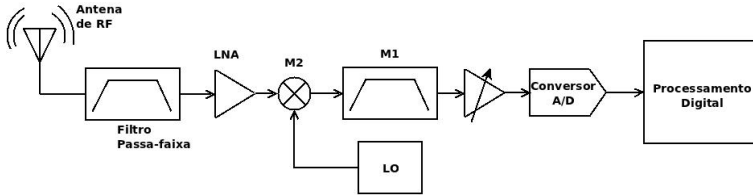


Figura 5 – Exemplo de digitalização de um sinal de rádio usando heteródino.

amplificador de baixo ruído (*Low Noise Amplifier* (LNA)) e um *mixer* multiplica o sinal por uma portadora gerada pelo Oscilador Local (*Local Oscillator* (LO)) para obter a IF (HAYKIN, 2008). Já na frequência intermediária (IF), o sinal é filtrado pelo filtro passa banda M1, amplificado e finalmente convertido para digital por um conversor A/D. No domínio digital, com o sinal sendo processado por um processador de sinal digital (*Digital Signal Processor* (DSP)) ou por um *Field-Programmable Gate Array* (FPGA) (representado pelo bloco **Processamento Digital** na Figura 5), um conversor digital (*Digital Down-converter* (DDC)) posiciona o espectro do sinal na banda base (ou frequência DC). É assumido que o LO pode ser programado para varrer a faixa de frequência de interesse. Na Figura 5, se um conversor A/D com frequência de amostragem $f_s = 96 \text{ MHz}$ é usado para digitalizar um sinal de RF que está originalmente na frequência de $1,8 \text{ GHz}$ e apresenta uma largura de banda de 20 MHz , o filtro M1, normalmente baseado em tecnologia de Ondas Acústicas de Superfície (*Surface Acoustic Wave* (SAW)), é responsável por limitar a largura de banda em 20 MHz . M1 assume que a frequência central do sinal em IF é pré-especificada (por exemplo 70 MHz) e frequentemente tem um fator **Q** (fator de qualidade) melhor do que o filtro de banda larga logo após a antena.

O LO deve sintonizar o sistema de forma que o sinal na saída do mixer M2 tenha o sinal dentro da faixa de interesse de M1. O conversor DDC, que é parte do bloco **Processamento Digital**, realiza uma "sintonia fina" da frequência central do sinal usando um oscilador numericamente controlado (*Numerically-controlled Oscillator* (NCO)). É assumido que todos esses elementos estão sob o controle do bloco AMC. Contudo, é mais comum encontrar situações em que os transmissores não são cooperativos e com isso, muitos parâmetros, como por exemplo a taxa de dados (*baud rate*), são desconhecidos. Este trabalho de mestrado assume duas abordagens. No capítulo 4 assume-se que o receptor conhece a taxa de dados do sinal (receptor sincronizado com o transmissor). O capítulo 5 apresenta uma análise aprofundada na qual assume-se que o receptor não conhece a taxa de dados do transmis-

sor. Para este último caso, o bloco AMC realiza uma varredura de frequência e identifica a banda de frequência que contém o sinal de interesse, o qual é digitalizado. Um modelo típico de AMC, apresentado em Dobre et al. (2007), assume que a classificação pode ser realizada em sequências de amostras do sinal recebido em banda base na saída de um filtro casado², o qual é definido como:

$$r_n = \alpha e^{j2\pi f_\theta T_n + j\theta_n s_n^{(y)} + g_n} \quad (2.1)$$

onde α é a amplitude do sinal, n é o índice do símbolo³, T é o período do símbolo, $S_n^{(y)}$ é o símbolo do y -ésimo tipo de modulação transmitido no tempo nT , f_θ é o *offset* de frequência⁴ e θ_n é o *jitter* de fase⁵.

O ruído complexo Gaussiano aditivo⁶ g_n tem média zero com variância igual a $\sigma_g^2/2$. Os símbolos $S_n^{(y)}$ são assumidos serem independentes e identicamente distribuídos, com valores dados pelo alfabeto do tipo da y -ésima modulação (MULLER; CARDOSO; KLAUTAU, 2011).

Para o caso em que não se considera o sincronismo entre transmissor e receptor (apresentado no capítulo 5), assume-se que r_n na equação 2.1 é obtido na taxa de $1/(T + \Delta T)$ onde $\Delta T \neq 0$. Já para o caso em que se considera o sincronismo do sinal (capítulo 4), o sinal r_n na equação 2.1 é obtido na taxa de $1/(T + \Delta T)$ fazendo $\Delta T = 0$.

Para o modelo adotado, o sinal transmitido é dado por:

$$s(t) = \sum_{n=-\infty}^{\infty} s_n^{(y)} p(t - nT), \quad (2.2)$$

²Filtro casado é um método utilizado para maximizar a *Signal-to-Noise Ratio* (SNR) de um sinal desejado, em uma determinada faixa de frequência, de forma que a parcela desejada do sinal (faixa de frequência que se quer discriminar) é separado mais facilmente da parcela indesejada (ruído misturado com o sinal recebido) (HAYKIN, 2008).

³A informação a ser transmitida é agrupada em conjuntos de bits e estes, por sua vez, compõem um símbolo que nada mais é do que uma forma de onda específica a ser transmitida em RF (SKLAR, 2001).

⁴Diferença entre a frequência do oscilador do transmissor e a frequência do oscilador do receptor (PROAKIS, 2008).

⁵Variação no período do sinal em relação a uma dada referência (PROAKIS, 2008).

⁶Modelo de ruído utilizado em teoria da informação para descrever os processos naturais que afetam a transmissão. É definido por um processo Gaussiano $n(t)$ randômico no qual o valor de n em qualquer tempo arbitrário t é caracterizado estatisticamente pela função de densidade Gaussiana:

$$p(n) = \frac{1}{(\sigma)\sqrt{2\pi}} \exp \left[-\frac{1}{2} \left(\frac{n}{\sigma} \right)^2 \right]$$

onde σ^2 é a variância de n . A função Gaussiana de média zero é obtida assumindo $\sigma = 1$. O termo "aditivo" significa que o ruído simplesmente é adicionado ao sinal desejado.

onde $p(t)$ é o pulso formatador.

O sinal recebido é $s(t) + g(t)$, onde $g(t)$ é o ruído branco Gaussiano aditivo⁷ (*Additive White Gaussian Noise* (AWGN)) ("ruído de canal" na Figura 2). Após ser filtrado por um filtro receptor com resposta ao impulso $f(t)$ e amostrado em uma taxa $1/(T + \Delta T)$, a informação passada para o módulo AMC é:

$$r_m = [(s(t) + g(t)) * f(t)]|_{t=m(T + \Delta T)}, \quad (2.3)$$

onde $*$ significa convolução⁸ e m indica que para $\Delta T \neq 0$, o número de símbolos no transmissor durante um certo intervalo de tempo será diferente do número de símbolos no receptor.

Comparando com a equação (2.1), o modelo adotado equivale a assumir uma recuperação perfeita da portadora no receptor, isto é, $f_\theta = 0$ e $\theta_n = 0$. Mas enquanto a equação (2.1) assume $\Delta T = 0$, como na maioria dos trabalhos em AMC, no capítulo 5 trabalha-se com erro na estimação da taxa de dados. Este fato tem importância prática, uma vez que AMC tipicamente trabalha com sistemas não cooperativos e os métodos de estimação de taxa de dados são "cegos" (*blind estimation*) e propensos a erros.

As Figuras 6 e 7 ilustram a diferença entre assumir $\Delta T = 0$ ou não. A Figura 6 mostra o diagrama de constelação de seis diferentes modulações (PAM4, PSK4, QAM4, PSK8, QAM16 e QAM64). A relação sinal-ruído (SNR) é ajustada como infinita e assume-se que o receptor faz amostragem do sinal na taxa de dados em sincronia com o transmissor. Na Figura 7, a modulação (PSK8) é escolhida para verificar o que ocorre com o diagrama de constelação quando o receptor amostra em taxas diferentes da taxa de dados do transmissor. Além disso, é introduzido um *offset* de fase de $\pi/2$ em relação à taxa de dados. Nos três diagramas superiores, o receptor amostra o sinal em uma taxa mais alta do que a taxa de dados⁹. No diagrama inferior esquerdo, o sinal é amostrado na taxa de dados do transmissor e é possível observar-se a influência do *offset* de fase. Pode-se comparar este diagrama com o diagrama

⁷O termo "branco" (*white*) é utilizado em uma analogia com a luz branca, que apresenta frequências em todo o espectro de cores. No caso de RF, significa que o ruído está igualmente distribuído em todas as frequências do sinal, independente do tempo (SKLAR, 2001).

⁸Operação matemática entre duas funções ($f(t)$ e $g(t)$) que gera uma terceira função, sendo definida pela seguinte integral (referenciada por integral de convolução):

$$f(t) * g(t) = \int_{-\infty}^{\infty} g(\lambda) f(t - \lambda) d\lambda$$

Em processamento de sinais, é utilizada para determinar a resposta que um sistema linear invariante no tempo (caracterizado por sua resposta ao impulso ($f(t)$)) apresenta quando em sua entrada está presente um determinado sinal ($g(t)$) (SKLAR, 2001).

⁹O valor 80 corresponde ao número de pontos que a ferramenta Matlab[®] necessita para representar um sinal como sendo contínuo.

de PSK8 na Figura 6. Um comportamento similar pode ser observado no diagrama inferior central, no qual um fator de subamostragem (*downsampling*) de $2/3$ é aplicado. No último diagrama (inferior direito), uma subamostragem também é aplicada, mas a constelação é mais parecida com os diagramas sobreamostrados da linha superior.

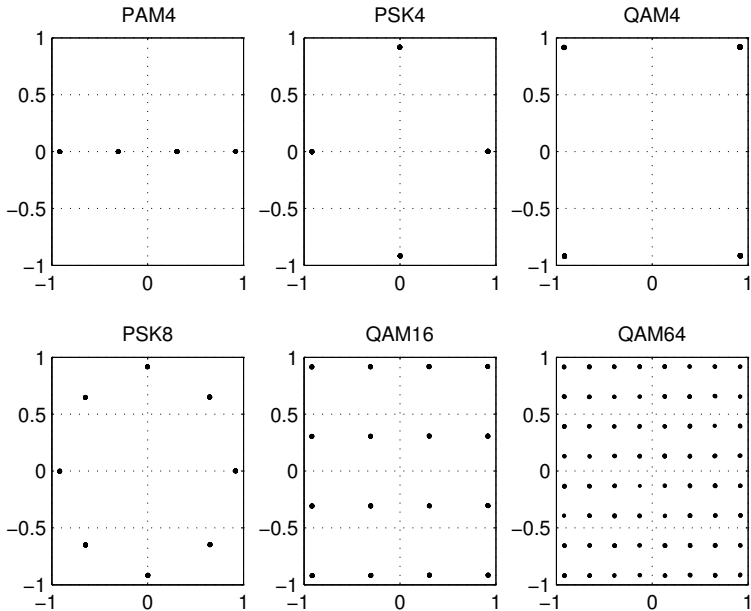


Figura 6 – Diagrama de constelação de modulações PAM4, PSK4, QAM4, PSK8, QAM16 e QAM64 em condições ideais.

É importante observar que a maioria dos trabalhos de AMC encontrados na literatura assume que a taxa de transmissão de dados é conhecida e é a mesma para o emissor e para o receptor. Entretanto, na prática é muito pouco provável que o receptor tenha conhecimento da taxa de transmissão, frequências de portadora ou mesmo do *offset* de frequências do sinal. Neste sentido, a análise realizada no capítulo 5 se diferencia dos trabalhos prévios, uma vez que o método de classificação adotado foi cuidadosamente investigado quanto a robustez, sem se assumir, *a priori*, uma taxa de transmissão específica, mas sim, uma faixa de possíveis taxas de transmissão.

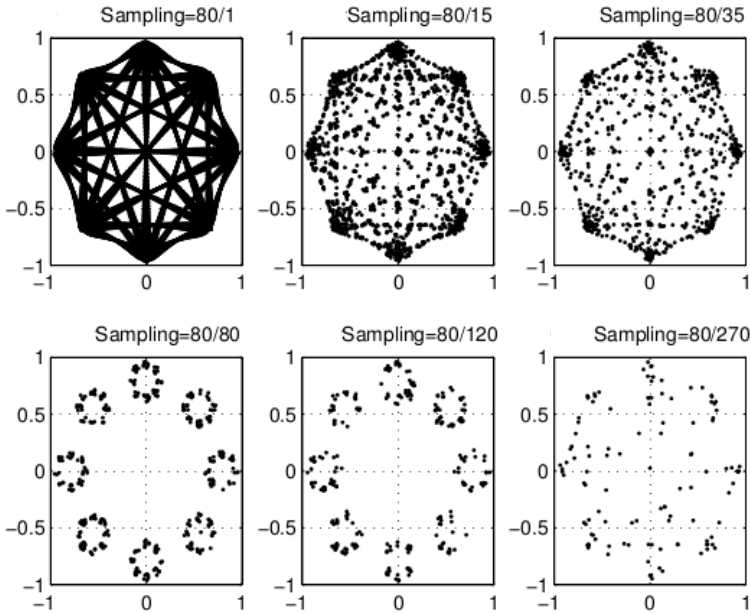


Figura 7 – Diagramas de constelação da modulação PSK8 com diferentes taxas de amostragem. A SNR é ajustada para infinito e um *offset* de fase de $\pi/2$ é aplicado.

2.2 CLASSIFICAÇÃO AUTOMÁTICA DE MODULAÇÕES (AMC)

A classificação automática de modulação (AMC) é um passo intermediário para a completa detecção de sinais de RF. Durante muito tempo o principal interesse de realizar classificação de modulações foi para fins militares, como uma forma de decodificar sinais de transmissões inimigas. Atualmente, no entanto, o rádio cognitivo (*Cognitive Radio* (CR)) trouxe uma nova aplicação para a AMC. Como uma das partes mais importantes em um sistema de CR, a AMC tem sido empregada para detectar e identificar a presença de usuários primários antes que estes sejam afetados pelas transmissões dos rádios cognitivos. Uma outra aplicação para AMC no contexto de CR é a de permitir comunicação entre diferentes equipamentos de CR (que podem operar em diferentes modulações) com o objetivo de viabilizar a interação entre eles (para monitoramento colaborativo do ambiente de RF, por exemplo). O principal desafio de AMC é atingir alta taxa de acertos na classificação de um sinal mesmo quando o intervalo de observação é curto e a SNR varia grande-

mente. Além disso, a AMC deve ser capaz de reconhecer uma grande quantidade de modulações diferentes em ambientes com condições de propagação diversas, possuir características de tempo real e baixa complexidade computacional (DOBRE et al., 2007). Outro desafio, já mencionado anteriormente, é o fato de que o sistema de AMC, na prática, dificilmente tem conhecimento da taxa de dados com que o usuário primário está transmitindo e tampouco possui dados suficientes para realizar qualquer espécie de sincronismo prévio. A identificação do tipo de modulação em tal condição representa um desafio muito maior do que o caso em que se dispõe de algum conhecimento prévio sobre o sinal que se quer identificar.

A figura 8 apresenta o diagrama em blocos genérico de um sistema de comunicação que realiza AMC na recepção.

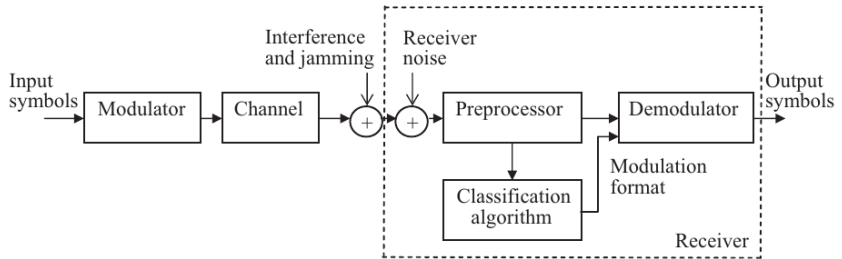


Figura 8 – Diagrama em blocos de AMC (DOBRE et al., 2007)

Um ponto importante a ser observado é que o sinal a ser processado pode sofrer interferências (intencionais ou acidentais) de outros transmissores (apresentado como *interference and jamming* na Figura 8), degradação devido ao ruído do canal e do receptor (*Receiver noise* na Figura 8). O sinal, então, deve ser pré-processado antes de ser enviado para o bloco que faz a classificação. A "qualidade" do pré-processamento é fortemente acoplada ao método de classificação utilizado. Quanto mais exigente for o método de classificação, em termos da taxa de acertos dos dados estimados, maior o trabalho demandado do bloco de pré-processamento¹⁰. Após a classificação, o resultado é usado pelo bloco demodulador para realizar a demodulação do sinal, no caso de se desejar interceptar o sinal.

Conforme já apresentado na subseção 1.4.1, existem dois métodos básicos para realizar a classificação de modulação:

1. Baseado em verossimilhança;

¹⁰O bloco *Preprocessor* da Figura 8 é equivalente ao "Bloco extrator de características" da Figura 2.

2. Baseado em características.

O método baseado em verossimilhança se subdivide em outros três grupos:

1. *Average Likelihood Ratio Test* (ALRT);
2. *Generalized Likelihood Ratio Test* (GLRT);
3. *Hybrid Likelihood Ratio Test* (HLRT).

A ideia básica por trás do classificador baseado em verossimilhança é que toda a informação necessária para a classificação pode ser obtida da função de densidade de probabilidade (*Probability Density Function* (PDF)) inerente do sinal modulado, observada na forma de onda recebida (DOBRE et al., 2007). Em geral, a determinação da função de verossimilhança (*Likelihood Function* (LF)), derivada da função PDF, tem alto custo computacional por envolver cálculos de integrais, somatórios, produtórios, somatórios de produtórios, etc. Além disso, sabe-se que os métodos baseados em ALRT necessitam conhecer previamente a PDF do sinal, o que pode ser impraticável. Os métodos baseados em GLRT sofrem do problema de constelações aninhadas (por exemplo QAM4, QAM8, QAM16 ...) e os métodos HLRT demoram muito tempo para chegar à solução de máxima verossimilhança (*maximum likelihood*) quando o número de variáveis desconhecidas é muito grande. Em resumo, o método baseado em verossimilhança, apesar de ser considerado ótimo no sentido Bayseano (minimiza a probabilidade de falso positivo), sofre do problema de alta complexidade computacional devido à dificuldade de encontrar a solução analítica exata quando o número de parâmetros aumenta (DOBRE et al., 2007). Este fato é especialmente notável quando se busca uma implementação em hardware.

Os classificadores baseados em características, por outro lado, fornecem uma solução subótima. Porém, devido à variedade possível de características que podem ser extraídas do sinal, o uso deste método conduz a implementações mais simples do que aquelas que seriam obtidas caso se adotasse um método baseado em verossimilhança.

Os classificadores baseados em características necessitam de um bloco anterior que faça a extração de parâmetros. Desta forma, o bloco AMC é subdividido em um bloco extrator de características e um bloco classificador, conforme apresentado na Figura 2.

Na literatura, as características mais comumente utilizadas para realizar a classificação são aquelas que podem ser extraídas da amplitude instantânea, da fase ou da frequência do sinal. Assim, a escolha das características a serem utilizadas depende do sinal. Por exemplo (DOBRE et al., 2007):

- Sinais modulados em *Frequency-shift Keying* (FSK) têm amplitude constante, sinais modulados em *Amplitude-shift Keying* (ASK) têm amplitude variável, sinais modulados em *Phase-shift Keying* (PSK) carregam informação na fase do sinal. Assim, uma *Discrete Fourier Transform* (DFT) da amplitude instantânea pode ser utilizada para fazer a discriminação entre essas classes de sinais;
- Sinais modulados em ASK e em *Binary Phase-shift Keying* (BPSK) não carregam informação na fase absoluta, ao passo que sinais modulados em *M-ary Phase-shift Keying* (M-PSK) (com $M > 2$) apresentam informação na fase absoluta. Desta forma, a variância de fase pode ser utilizada para discriminar entre esses dois conjuntos de classes de sinais;
- Semelhantemente, outras características podem ser extraídas para sinais *Quadrature Amplitude Modulation* (QAM), PSK, *Pulse-amplitude Modulation* (PAM), etc.

Para cada grupo de características extraídas, um método de classificação pode ser utilizado de modo que, para classificar uma gama de modulações com diferentes características, uma árvore de decisão binária poderia ser utilizada para reunir todos os métodos baseados em características. Cada característica extraída seria comparada com um limiar em cada nó da árvore. No entanto, um sistema de tal ordem de complexidade demandaria um grande esforço computacional devido ao tratamento diferenciado que cada característica extraída de cada modulação iria requerer.

Ainda assim, a escolha mais óbvia para uma implementação em hardware é o método baseado em características, devido ao menor custo computacional quando, comparado com o método baseado em verossimilhança.

Entre os possíveis métodos de classificação utilizando características do sinal, as Máquinas de Vetores de Suporte (SVM) tem surgido como uma opção atrativa devido a sua boa performance e habilidade de lidar com grande quantidade de dados com um custo computacional aceitável.

A próxima seção traz uma introdução às Máquinas de Vetores de Suporte (SVM), apresentando os conceitos mais relevantes para a compreensão do restante do presente texto. Um estudo aprofundado das SVMs e de suas origens é encontrado em (VAPNIK, 1999). Uma descrição concisa, porém bastante completa, é encontrada em (BURGES, 1998).

2.3 INTRODUÇÃO ÀS MÁQUINAS DE VETOR DE SUPORTE - SUPPORT VECTOR MACHINES (SVM)

No contexto de máquinas de aprendizado, a SVM é uma **máquina de aprendizado supervisionado para classificação** (SMOLA; SCHÖLKOPF, 2004), (HEARST et al., 1998).

Usando um conjunto de entradas de teste (também referenciado por "conjunto de teste") e um rótulo para cada classe que se deseja classificar, a SVM "aprende" a prever os rótulos para novos conjuntos de entradas. É importante ressaltar que uma SVM padrão é um classificador binário, uma vez que somente dois rótulos são permitidos. Esse fato tem uma consequência prática sobre a implementação das arquiteturas dos classificadores, que será abordada no decorrer deste trabalho de mestrado.

Em classificação de dados, duas situações são encontradas: quando os dados podem ser separados por uma função linear e quando os dados não podem ser separados por uma função linear. As duas próximas subseções abordam esses dois casos.

2.3.1 Caso Linearmente Separável

Seja $\mathbf{x}_i \in \mathcal{X}^n$ um vetor de entrada n -dimensional ao qual é associado o rótulo $y_i \in \{-1, 1\}$. Na fase de treino, os rótulos y_i dos respectivos x_i são conhecidos. Na fase de teste, dadas x_i entradas, a SVM fornece y_i saídas.

Na fase de treino são considerados M vetores de entrada como sendo M pontos no hiperespaço. Após, supõe-se que existe um hiperplano que separa os pontos em dois grupos de acordo com seus rótulos. Isso pode ser expresso como M desigualdades da forma:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) > 0, \quad (2.4)$$

onde \mathbf{w} é o vetor que define a orientação do hiperplano e b é o *offset* do hiperplano, também chamado de limiar (*threshold*). Note que a equação 2.4 somente fornece um valor válido para o problema em questão porque os rótulos y_i são definidos como tendo sinais opostos.

A SVM encontra os valores de \mathbf{w} e b que definem o hiperplano de separação com a maior distância entre os pontos mais próximos de cada grupo. Esta distância é igual a $1/\|\mathbf{w}\|$ (veja (BURGES, 1998) para maiores detalhes). A Figura 9 mostra duas classes (círculos e pontos), o hiperplano de separação (a linha cheia entre as duas linhas pontilhadas), a margem de separação (*margin*) e os vetores de suporte (H_1 e H_2), além de b e \mathbf{w} .

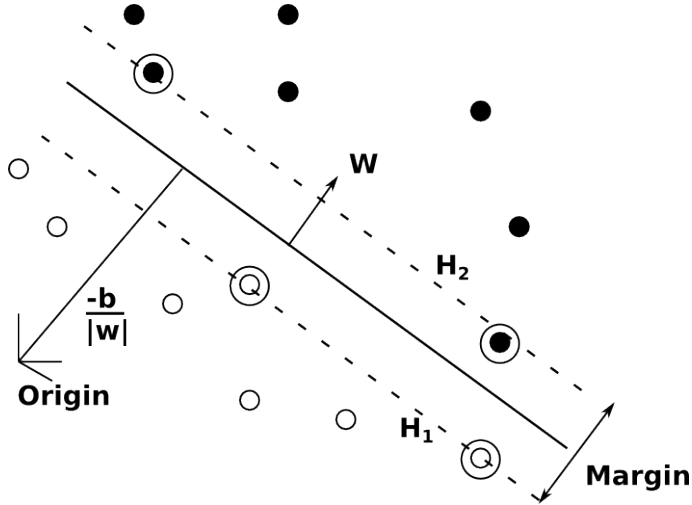


Figura 9 – Conjunto de dados linearmente separável (BURGES, 1998)

Assim, a fase de treinamento pode ser interpretada como um problema de otimização¹¹:

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{sujeito a} \quad & : y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \end{aligned} \quad (2.5)$$

Usualmente, é mais conveniente introduzir **Multiplicadores de Lagrange** e resolver o problema dual¹¹. A cada restrição apresentada na Equação 2.5 é associado um Multiplicador de Lagrange. Após algumas manipulações algébricas o problema de otimização torna-se:

¹¹Em programação linear, qualquer problema tem associado outro problema chamado dual. Neste contexto, o problema original denomina-se primal. Dentre as regras que regem a transformação do problema primal para dual está aquela que se o problema primal de forma típica for de maximização, o problema dual de forma típica será de minimização. A aplicação do método permite determinar um valor ótimo que resolve o problema primal original e o problema dual (CORMEN et al., 2009).

$$\begin{aligned}
& \max_{\alpha} \quad \sum_{i=1}^M \alpha_i - \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^M \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\
& \text{sujeito a} \quad : \quad \alpha_i \geq 0 \\
& \quad \quad \quad \sum_{i=1}^M y_i \alpha_i = 0
\end{aligned} \tag{2.6}$$

A formulação anterior é resolvida usando métodos de programação quadrática. O hiperplano do problema primal é relacionado com o problema dual por:

$$\mathbf{w}^* = \sum_{i=1}^M \alpha_i^* y_i \mathbf{x}_i \tag{2.7}$$

onde o asterisco faz referência ao valor ótimo encontrado na otimização¹¹. Os vetores de suporte são os pontos \mathbf{x}_i para os quais os Multiplicadores de Lagrange α_i^* são estritamente maiores do que zero. Na fase de teste, os valores ótimos \mathbf{w}^* , b^* são usados na seguinte função para prever o rótulo $y \in \{-1, 1\}$ de um vetor de entrada \mathbf{x} :

$$y = f(\mathbf{x}) = \text{sign}(\mathbf{w}^* \cdot \mathbf{x} + b^*). \tag{2.8}$$

2.3.2 Caso Não Separável Linearmente

Na subseção 2.3.1 foi suposto que os dados são linearmente separáveis. Isto significa que existe um hiperplano que divide os pontos de treino em dois grupos, com seus respectivos rótulos. Mas o caso mais comumente encontrado é aquele em que os dados não são separáveis. Para lidar com essa situação são introduzidas as variáveis de relaxação (*slack variables*) $\xi_i \geq 0$ nas restrições da equação 2.5. Desta forma, o problema de otimização primal torna-se:

$$\begin{aligned}
& \min_{\mathbf{w}, b} \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^M \xi_i \\
& \text{sujeito a} \quad : \quad y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i
\end{aligned} \tag{2.9}$$

onde o coeficiente C é um parâmetro de treinamento definido pelo usuário. Se o valor de C é grande, uma alta penalidade está associada com as variáveis

de relaxação. Se o conjunto de treino é linearmente separável, este coeficiente é menos importante. Esta modificação no problema primal causa uma pequena modificação no problema dual: os Multiplicadores de Lagrange são restringidos para o intervalo $0 \leq \alpha_i \leq C$. Isto implica que os coeficientes armazenados na memória têm uma faixa limitada. Na Figura 10 pode-se observar um conjunto de dados que não são separáveis linearmente. Nota-se a presença de amostras da classe "ponto" (classe 1) abaixo do hiperplano de separação (região pertencente à classe "círculo" (classe -1)) e vice-versa, bem como a variável de relaxação ξ . Na Figura 10 também aparecem b e w .

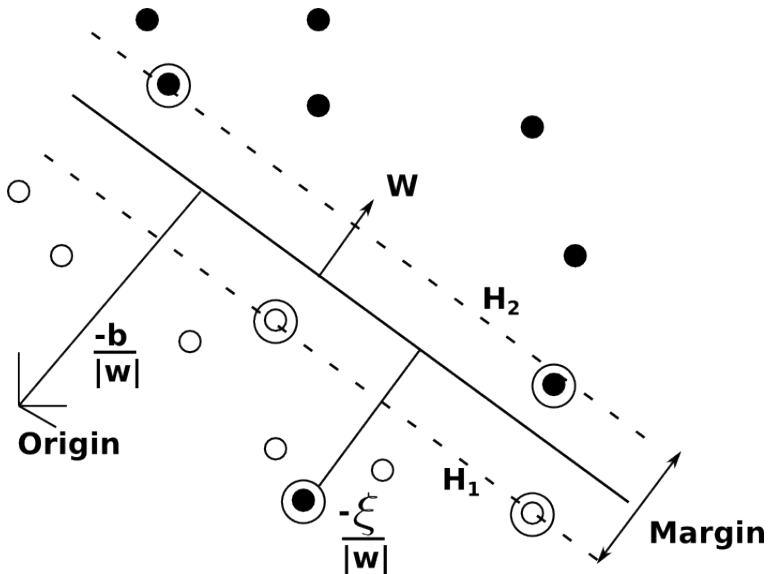


Figura 10 – Conjunto de dados não separáveis linearmente (BURGES, 1998)

2.3.3 Combinando Extrator de Parâmetros Baseado em Histograma e SVM Binária Linear para Realizar AMC

Nesta subseção é mostrado que a função de predição da equação 2.8 pode ser eficientemente computada sem multiplicações se o vetor de entrada x for o histograma do sinal.

Um histograma é uma representação gráfica de um conjunto de dados que podem ser agrupados em classes. O método de representação é um retângulo que tem por base horizontal as classes e a altura vertical representa

o número de vezes (frequência) que a classe aparece no conjunto de dados. No contexto deste trabalho, a base (dimensão horizontal) de cada retângulo (referenciados por *bins*) representa uma parcela da amostra complexa (dividida em parte real e imaginária ou ainda, na forma polar, em módulo e fase) e a altura (dimensão vertical) representa o número de vezes (frequência) que essa amostra aparece no sinal amostrado. O histograma completo (conjunto de bins) forma a assinatura da modulação.

Considerando um histograma de K bins de forma que $\mathbf{x} = (x_1, \dots, x_K)$ e $\mathbf{w} = (w_1, \dots, w_K)$, a equação 2.8 pode ser escrita na forma:

$$f(\mathbf{x}) = \text{sign}\left(\sum_{j=1}^K w_j x_j + b\right) \quad (2.10)$$

onde x_j é o número de ocorrências no bin j . Uma vez que x_j é um inteiro positivo, a equação 2.10 pode ser reescrita como:

$$f(\mathbf{x}) = \text{sign}\left(\sum_{j=1}^K \sum_{l=1}^{x_j} w_j + b\right) \quad (2.11)$$

Em um primeiro momento, pode parecer que fazer sucessivas somas de w_j não é mais eficiente do que fazer as multiplicações $w_j x_j$. Porém, em uma aplicação em tempo real, as amostras chegam uma após a outra. Desta forma, usando a equação 2.11 não é necessário computar x_j (o histograma). A soma pode ser realizada enquanto as amostras chegam. Após N amostras, é adicionado o valor b e o sinal é classificado.

2.3.4 O Problema de Classificação de Múltiplas Classes usando SVM

O algoritmo de SVM proposto por Cortes e Vapnik (1995) tem capacidade de discriminar uma entre duas classes. No entanto, a grande maioria dos problemas práticos apresenta mais de uma classe a ser discriminada. Isso dá origem ao problema de classificação de múltiplas classes, o qual pode ser resolvido de duas formas:

- Alterar o algoritmo da SVM de forma a considerar todas as classes de uma só vez;
- Dividir o problema em diversos classificadores binários e fazer combinação dos resultados.

O custo computacional para resolver o problema de otimização implícito no primeiro caso é muito maior do que o custo computacional de imple-

mentar múltiplas SVMs e combiná-las. Por esse motivo, a abordagem mais encontrada na literatura para resolver o problema de classificação de múltiplas classes é a decomposição do problema de forma a possibilitar a aplicação de um conjunto de SVMs binárias para sua classificação (HSU; LIN, 2002).

Hsu e Lin (2002) fizeram uma análise de três métodos que são os mais amplamente utilizados para a tarefa de decomposição:

1. Um contra todos - (*One Against All* (OAA));
2. Um contra um - (*One Against One* (OAO));
3. SVM de grafo acíclico direcionado - (*Directed Acyclic Graph SVM* (DAGSVM)).

O método OAA é, provavelmente, o mais antigo a ser usado para resolver o problema de classificação de múltiplas classes com SVM. Neste método, a classe que se quer identificar recebe um rótulo (*label*) positivo durante o treino da SVM, ao passo que as demais classes recebem um rótulo negativo. O mesmo procedimento é repetido para todas as demais classes, sempre fazendo com que a classe de interesse receba um rótulo positivo, ao passo que as demais recebem um rótulo negativo. O número de SVMs necessárias para classificar k classes é igual ao número de classes ($y = k$, onde y é o número de SVMs e k é o número de classes).

No caso do método OAO, as SVMs são treinadas para discriminar entre duas classes. Após a etapa de classificação, um esquema de decisão baseado em contagem (*score*) (votador), é utilizado para definir a classe que recebeu o maior número de votos. Nesse método, o número de SVMs necessárias para realizar a classificação é dado por:

$$y = k(k - 1)/2 \quad (2.12)$$

O método DAGSVM utiliza o mesmo esquema de OAO na fase de treino (portanto requer $k(k - 1)/2$ SVMs) e na fase de teste utiliza um grafo acíclico direcionado (DAGSVM) com $k(k - 1)/2$ nós internos e k folhas. Cada nó possui uma SVM binária que discrimina entre a i -ésima e a j -ésima classes. Partindo do nó raiz, a árvore é percorrida até a folha que contém a predição da classe.

Hsu e Lin (2002) fizeram a implementação dos três métodos para a fase de testes, alterando o algoritmo do software *LIBSVM* (CHANG; LIN, 2011). A taxa de acertos foi utilizada como principal métrica de comparação entre estes três métodos. Como métricas secundárias, os autores utilizam também os tempos de treino e classificação. Como uma forma de ter uma referên-

cia para comparação, Hsu e Lin (2002) utilizaram duas bases de dados bastante conhecidas para avaliação do problema de classificação para máquinas de aprendizado, Statlog collection (MICHIE; SPIEGELHALTER; TAYLOR, 1994) e UCI repository (BACHE; LICHMAN, 2013), as quais contêm problemas clássicos de classificação (HSU; LIN, 2002). Os resultados apresentados por esses autores mostraram que, em termos de taxa de acertos, os três métodos entregam resultados muito parecidos. Por outro lado, para cada tipo de teste, os três métodos se diferem bastante em termos de tempo de treino e tempo de teste (classificação). Contudo, para todos os casos, o método OAA apresenta tempo de treino maior que os outros dois métodos, chegando, em alguns casos, a apresentar uma ordem de grandeza de diferença nos tempos absolutos de treino. O tempo de teste (classificação), quando avaliado, não apresentou diferença significativa.

O próximo capítulo revisa os principais trabalhos na área de AMC, com ênfase para aqueles que utilizam SVM como classificador.

3 REVISÃO DE TRABALHOS CORRELATOS

Este capítulo apresenta uma visão geral dos principais trabalhos de classificação automática de modulação (*Automatic Modulation Classification* (AMC)) encontrados na literatura, com ênfase em *Support Vector Machines* (SVMs).

Cabe ressaltar que existem realmente poucos trabalhos correlatos, quando se considera o escopo do método aqui proposto, o qual conjuga a utilização de histogramas e SVM para construir os classificadores visando sua implementação em hardware dedicado. Portanto, a análise dos trabalhos correlatos apresentados neste capítulo objetivou uma avaliação genérica dos métodos por esses empregados. Tal avaliação leva em conta o tipo de parâmetro extraído para a classificação e tenta estimar a complexidade computacional envolvida para implementar tais métodos em hardware dedicado.

Tal condição impede uma comparação direta de valores de síntese, quando apresentado, seja para *Field-Programmable Gate Array* (FPGA) seja para *standard cells*.

3.1 CLASSIFICADOR BASEADO NA FUNÇÃO DO TESTE DA TAXA MÉDIA DE VEROSSIMILHANÇA (ALRT)

Os classificadores baseados em *Average Likelihood Ratio Test* (ALRT) são amplamente utilizados em algoritmos para AMC e são considerados os classificadores que fornecem o melhor resultado de classificação. Por esse motivo, optou-se por apresentar uma implementação de tal tipo de classificador.

No classificador apresentado por Su, Xu e Zhou (2008), um sinal R_n recebido, corrompido por um ruído, é tratado como uma variável randômica. Essa variável é então comparada com M_i símbolos previamente conhecidos.

A classificação é realizada utilizando a função de teste de verossimilhança, a qual pode ser obtida usando a função de densidade de probabilidade conjunta (*Probability Density Function* (PDF)) dada por:

$$L(R|H_i) = \sum_{n=1}^N T^i(R_n) \quad (3.1)$$

onde,

$$T^i(R_n) = \ln\left\{\frac{1}{2\pi\sigma^2 M_i} \sum_{k=1}^{M_i} \exp\left[-\frac{|R_n - S_k^i|^2}{2\sigma^2}\right]\right\} \quad (3.2)$$

R_n = Sequência de dados complexos de comprimento N , recebidos em banda base;

σ^2 = Variância do ruído Gaussiano aditivo (*Additive White Gaussian Noise* (AWGN)) correspondente a uma certa relação sinal-ruído (*Signal-to-Noise Ratio* (SNR));

M_i = Número de símbolos na i -ésima constelação;

S_k^i = k -ésimo símbolo de referência da i -ésima constelação;

$k = 1, 2, \dots, M_i$.

A classificação ocorre pela escolha da modulação para a qual $L(R|H_i)$ é máxima.

O desempenho do método de classificação baseado em ALRT depende de diversos parâmetros, como por exemplo, a variância do ruído, frequência e fase da portadora, taxa de símbolos e fator de forma do pulso.

Su, Xu e Zhou (2008) valem-se do fato de que em rádio definido por software (*Software-Defined Radio* (SDR)) o universo de modulações a serem identificadas é conhecido e portanto, a função discreta do teste da taxa média de verossimilhança (*Discrete Likelihood Ratio Test* (DLRT)) de cada modulação pode ser pré-calculada para uma dada gama de variação de SNR. Para cada modulação e para cada valor de SNR, as amostras complexas do sinal são separadas em componentes em fase e em quadratura e armazenadas em uma matriz $[x_p \ X \ y_q]$, onde x_p é a componente em fase e y_q é a componente em quadratura do sinal utilizado no pré-cálculo da função de DLRT. Esses valores de DLRT são armazenados em *Look-up Table* (LUT) para posteriormente serem comparadas com o sinal recebido R_n .

Apesar de não realizar operações de multiplicação e exponenciação, o método proposto em Su, Xu e Zhou (2008) exige espaço para armazenar os valores pré-calculados da função de DLRT.

Assumindo que cada amostra complexa é composta de uma componente em fase e outra componente em quadratura, e que cada parte é armazenada em 16 bits, para cada SNR serão necessários 160 *kbits* para armazenar 50 amostras complexas para 4 modulações diferentes $(16 \times 50 \times 50 \times 4)^1$. Se considerarmos 16 valores de SNR, serão necessários 2,56 *Mbits*. Em uma implementação em hardware para este método, tal quantidade de memória requereria um bloco de memória com área e tempos de acesso não desprezíveis.

¹As amostras complexas são armazenadas em uma matriz x_p por y_q , componentes em fase e em quadratura (50X50 no exemplo).

3.2 CLASSIFICADORES BASEADOS EM MÁQUINAS DE VETORES DE SUPORTE (SUPPORT VECTOR MACHINES - SVM)

Conforme apresentado na seção 2.3, as máquinas de vetores de suporte vem sendo bastante utilizadas para realizar a tarefa de classificação.

Assim, as Subseções 3.2.1, 3.2.2, 3.2.3 e 3.2.4 apresentam os principais trabalhos em AMC que utilizam SVM com *kernel* não linear, ao passo que as Subseções 3.2.5 e 3.2.6 apresentam trabalhos que utilizam SVM com *kernel* linear.

3.2.1 Xiao et al. (2012)

Xiao et al. (2012) fazem uso de histogramas da frequência instantânea do sinal em conjunto com SVM para classificação de sinais analógicos. A extração dos parâmetros é realizada em duas etapas: estimação da frequência instantânea e cálculo do histograma da frequência instantânea estimada.

A frequência instantânea é obtida da diferença da fase instantânea de duas amostras complexas consecutivas. O histograma das diferenças de fase carrega informação de probabilidade de distribuição da frequência instantânea, a qual pode ser utilizada como assinatura do sinal em teste.

Nesse trabalho, os autores propõe o uso de SVM com *kernel* polinomial dado por:

$$k(x_i, x_j) = (\gamma x_i^T x_j + \lambda)^d, \gamma > 0 \quad (3.3)$$

onde γ , λ e d são parâmetros do *kernel*. O método *grid search* com validação cruzada, o qual fornece valores ótimos, foi utilizado para determinação destes parâmetros.

Os autores testaram as modulações *Amplitude Modulation* (AM), *Frequency Modulation* (FM), *Double Sideband* (DSB), *Lower Sideband* (LSB) e *Upper Sideband* (USB) em ambiente simulado (Matlab®) e em ambiente real (embora não informem qual a infraestrutura utilizada). Utilizando simulações, os autores determinaram que o algoritmo consegue realizar a classificação em um tempo médio de 100 ms com uma taxa de acertos da classificação acima de 93% para uma taxa de sinal-ruído(SNR) = 10 dB.

Embora os resultados apresentados sejam relevantes, tanto para o teste prático quanto para a simulação, os autores concentraram sua pesquisa em modulações analógicas, as quais atualmente tem seu maior uso em ambientes militares, o que limita bastante o escopo do trabalho.

Outro ponto a ser considerado é o fato de os autores utilizarem uma

SVM de *kernel* não linear. Apesar de citarem testes práticos, os autores não explicitaram o ambiente no qual estes testes foram realizados.

3.2.2 Tabatabaei, Krishnan e Anpalagan (2010)

Tabatabaei, Krishnan e Anpalagan (2010) utilizam quatro características do sinal em conjunto com classificadores baseados em SVM para classificar sinais modulados em *amplitude-shift keying* (ASK4), *phase-shift keying* (PSK8) e *frequency-shift keying* (FSK8).

As quatro características extraídas do sinal são:

- Entropia Renyi;
- Variação na frequência;
- Assimetria do espectro;
- Passagem por zero.

Essas características extraídas do sinal são combinadas em um vetor e aplicadas a uma SVM de *kernel* não linear. Dois *kernels* diferentes foram testados:

- Kernel *Radial Basis Function* (RBF) dado por:

$$K(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right) \quad (3.4)$$

onde x é o vetor de entrada, y são os rótulos e σ é o parâmetro de *kernel* RBF.

- *Multilayer Perceptron* (MLP) dado por:

$$K(x, y) = \tanh(kx^T + \theta) \quad (3.5)$$

As simulações realizadas pelos autores apresentaram resultados globais de classificação iguais a 55,2%, 67,7% e 77% para 10 dB, 15 dB e 20 dB, respectivamente, sendo que o *kernel* RBF apresentou melhores resultados.

A exemplo do trabalho de Xiao et al. (2012), o trabalho de Tabatabaei, Krishnan e Anpalagan (2010) também utiliza uma função de *kernel* não linear para a SVM.

Outro fator que também deve ser levado em consideração é o custo computacional para extração das características o qual, possivelmente, torna a implementação em hardware dedicado proibitiva.

3.2.3 Wu et al. (2005)

Wu et al. (2005) utilizaram uma SVM não linear como classificador para características extraídas da amplitude, fase e frequência instantâneas do sinal. Essas características são combinadas de forma a gerar um conjunto de dados que pode ser separado por uma SVM de *kernel* não linear RBF dada pela equação 3.4.

Para a abordagem proposta, o valor do parâmetro de *kernel* RBF σ é determinado através de um processo iterativo de seleção baseado no conjunto completo de características. Resultados simulados para as modulações ASK2, ASK4, FSK2, FSK4, BPSK e QPSK em uma SNR de 5 dB mostraram uma taxa de acertos da classificação mínima de 85,67% (obtida para a modulação ASK4).

3.2.4 Hu, Song e Wang (2008)

Hu, Song e Wang (2008) propõem o uso de um conjunto de quatro parâmetros característicos de coerência espectral, obtidos via análise de correlação espectral, e SVM para a classificação de sinais modulados em AM, ASK, FSK, PSK, MSK e QPSK. Os parâmetros utilizados são devidos a características cicloestacionárias do sinal modulado, geradas por operações de amostragem, modulação, multiplexação e codificação do sinal. Para realizar a análise de cicloestacionaridade do sinal, duas funções são normalmente utilizadas: a função de autocorrelação cíclica (*Cyclic Autocorrelation Function* (CAF)), apresentada na Equação 3.6, e a função de correlação espectral (*Spectral Correlation Function* (SCF)), apresentada na Equação 3.7. Após manipulação algébrica (3.6 em 3.7) obtêm-se a equação 3.9.

$$R_x^\alpha(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{\frac{T}{2}}^{\frac{-T}{2}} x(t + \frac{\tau}{2}) x(t - \frac{\tau}{2}) e^{-j2\pi\alpha t} dt \quad (3.6)$$

$$S_x^\alpha(f) = \int_{-\infty}^{+\infty} R_x^\alpha(\tau) e^{-j2\pi f \tau} d\tau \quad (3.7)$$

$$= \int_{-\infty}^{+\infty} \lim_{T \rightarrow \infty} \frac{1}{T} \int_{\frac{T}{2}}^{\frac{-T}{2}} x(t + \frac{\tau}{2}) x(t - \frac{\tau}{2}) e^{-j2\pi\alpha t} e^{-j2\pi f \tau} dt d\tau \quad (3.8)$$

$$= \lim_{T \rightarrow \infty} S_{xT}^\alpha(t, f) \quad (3.9)$$

onde $x(t)$ é o sinal no domínio do tempo, α é a frequência do ciclo, f o espectro de frequência e $S_{xT}^\alpha(t, f)$ é o periodograma cíclico de $S_x^\alpha(f)$. A CAF possibilita uma análise no domínio do tempo e a SCF possibilita uma análise espectral da correlação do sinal $x(t)$.

Na prática, o número de amostras é limitado e por esse motivo, a função precisa ser estimada. Os autores optaram por utilizar um método chamado *frequency-smoothing* para essa tarefa. Com isso, obtiveram as seguintes características do sinal:

- Número de pulsos em f ;
- Número de linhas espectrais;
- Média de energia nas linhas espectrais;
- Valor máximo do coeficiente de coerência espectral.

Estes parâmetros são então utilizados como entradas para o bloco classificador (SVM).

Os resultados de simulação apresentados pelos autores mostram uma taxa de acertos da classificação acima de 85,1% (obtida para a modulação QPSK).

Os trabalhos propostos por Xiao et al. (2012), Tabatabaei, Krishnan e Anpalagan (2010), Wu et al. (2005) e Hu, Song e Wang (2008) fazem uso de um *kernel* não linear (na maioria deles o RBF). É razoável admitir-se que a implementação de uma SVM com *kernel* não linear apresente um custo de hardware superior àquele de uma SVM de *kernel* linear, uma vez que SVMs com *kernel* não linear exigem funções mais complexas (como por exemplo polinômios, uma das funções de *kernel* não linear mais simples) do que aquelas exigidas por SVMs com *kernel* linear.

3.2.5 Muller, Cardoso e Klautau (2011)

Muller, Cardoso e Klautau (2011) apresentam um extrator de parâmetros baseado nas características do sinal, chamado de *Concatenated Sorted Symbols* (CSS) que, combinado com o classificador SVM, resulta em uma arquitetura passível de ser implementada em hardware dedicado a um baixo custo. O extrator de parâmetros CSS utiliza a amplitude e fase instantâneas do sinal para construir uma assinatura digital do mesmo. Basicamente, o sinal complexo amostrado em banda base tem suas partes real e imaginária separadas, individualmente ordenadas, e então concatenadas, formando a assinatura de cada modulação.

Os autores realizaram comparações da taxa de acertos do classificador SVM utilizando o extrator de parâmetros CSS (SVM-CSS) com métodos clássicos encontrados na literatura: o ALRT, o DLRT e outros dois classificadores baseados em classificação hierárquica. O conjunto de modulações utilizado para a experimentação foi BPSK, PAM4, QAM16 e PSK8 e a faixa de SNR em teste foi de -5 dB a 15 dB. Os resultados de taxa de acertos de classificação alcançados em simulações mostram que o SVM-CSS apresenta uma precisão que segue a curva do ALRT, considerada curva de referência, com uma taxa de acertos apenas 3 dB menor, no pior caso, para toda a faixa de SNR testada.

Contudo, o classificador CSS-SVM apresenta como ponto fraco o fato de precisar armazenar um certo número de amostras para então ordená-las e só então realizar a classificação. Esse ordenamento demanda espaço de memória e, conseqüentemente, aumenta o custo de hardware. Ainda assim, o método proposto apresenta custo computacional que é compatível com implementação em hardware dedicado.

3.2.6 Cardoso, Castro e Klautau (2013)

Um trabalho que apresenta resultados bem mais condizentes com implementações em hardware foi desenvolvido por Cardoso, Castro e Klautau (2013). Neste trabalho, os autores apresentam um extrator de parâmetros baseado em histograma que, combinado com um classificador SVM linear, resultou em uma arquitetura de hardware simples o suficiente para ser embarcado em um chip FPGA ou dedicado. Basicamente, os histogramas das partes real e imaginária das amostras complexas do sinal são extraídos e utilizados como uma assinatura para classificar a modulação. Este extrator de parâmetros foi batizado de *HISTO*.

Os autores realizaram testes com quatro modulações, BPSK, PAM4, QAM16 e PSK8 em uma faixa de SNR que varia de -5 dB até 10 dB. Os resultados apresentaram uma taxa de acertos próxima de 50% para uma SNR = -5 dB.

Para fins de comparações de desempenho e de custo de hardware, os autores fizeram a síntese do classificador baseado em HISTO (*HISTO-SVM*) e do classificador baseado em CSS (*CSS-SVM*) para o FPGA *Cyclone II EP2C20F484C* da Altera. Os resultados alcançados mostram taxa de acertos de classificação praticamente igual para os classificadores CSS-SVM e HISTO-SVM em uma faixa de -5 dB até 10 dB de SNR.

Vale observar que o classificador *HISTO-SVM* utiliza um histograma simples (*1-D*) o que resulta em uma perda de informação do sinal. Entre as

questões abordadas no presente trabalho de mestrado, busca-se eliminar tal perda de informação adotando-se o uso de histograma de duas dimensões.

3.3 CONCLUSÕES DO CAPÍTULO

A Tabela 1 a seguir sumariza os trabalhos discutidos nesta seção.

É importante ressaltar que é difícil estabelecer uma comparação detalhada e justa entre os trabalhos pois os resultados de taxa de acertos da classificação são fortemente dependentes do tipo de modulação adotada e do tipo de característica utilizada para a classificação.

Para o presente trabalho de mestrado, o principal interesse é identificar arquiteturas do conjunto extrator de parâmetros e classificador passíveis de implementação em hardware. Levando em conta esses quesitos, o trabalho apresentado por Cardoso, Castro e Klautau (2013) é a referência mais importante, tendo servido de inspiração para o presente trabalho. Neste sentido, os próximos dois capítulos apresentam os experimentos realizados, os quais buscaram aperfeiçoar o trabalho de Cardoso, Castro e Klautau (2013), a partir de uma análise aprofundada e com premissas mais realistas, no sentido de melhorar a taxa de acertos e de propor arquiteturas de hardware dedicadas otimizadas.

Tabela 1 – Tabela comparativa dos trabalhos apresentados

Autores	Modulações	Faixa de SNR (dB)	Kernel SVM	Taxa de acertos obtida (pior valor) %	Teste realizado
(SU; XU; ZHOU, 2008)	BPSK, QPSK, QAM16, QAM32, QAM64	0 a + 12	-	>60	simulação (Monte Carlo)
(XIAO et al., 2012)	AM, FM, DSB, LSB, USB	-4 a 14	Polinomial	1(USB)	simulação (Matlab®) & Teste prático
(TABATABAEI; KRISHNAN; ANPALAGAN, 2010)	ASK4, PSK8, FSK8	10,15,20	RBF; MLP	55,2	simulação (Matlab®)
(WU et al., 2005)	ASK2, ASK4, FSK2, FSK4, BPSK, QPSK	+5	RBF	85,67	simulação (Matlab®)
(HU; SONG; WANG, 2008)	AM, ASK, FSK, PSK, MSK, QPSK	0 a +20	RBF	85.1	simulação (Matlab®)
(MULLER; CARDOSO; KLAUTAU, 2011)	BPSK, PAM4, QAM16, PSK8	-5 a +15	Linear	>45	simulação (Matlab®)
(CARDOSO; CASTRO; KLAUTAU, 2013)	BPSK, PAM4, QAM16, PSK8	-5 a +10	Linear	≈ 47	simulação (Matlab®) & Teste prático

4 CLASSIFICADORES BASEADOS EM SVM BINÁRIA LINEAR E HISTOGRAMAS 1-D

Este capítulo apresenta os classificadores de modulações digitais baseados em *Support Vector Machine* (SVM) utilizados em conjunto com o extrator de parâmetros baseado em *Histogramas 1-D* (HISTO-1D), os quais serão referenciados por SVM-HISTO 1-D.

A seção 4.1 apresenta a infraestrutura utilizada para gerar os sinais modulados, treinar as SVMs e gerar os coeficientes a serem utilizados na etapa de classificação (teste¹) das modulações. Uma descrição detalhada dos *scripts* Matlab[®] utilizados é apresentada, objetivando facilitar o entendimento do restante do capítulo.

Na seção 4.2 o *datapath* de uma SVM linear utilizando o extrator de parâmetros HISTO-1D é apresentado e discutido.

As seções 4.3 e 4.4 apresentam, respectivamente, arquiteturas para os blocos classificadores SVM-HISTO 1-D que usam os esquemas de decisão de múltiplas classes *One Against One* (OAO) e *One Against All* (OAA). Estas arquiteturas foram sintetizadas em um fluxo *standard cell* com uma ferramenta industrial, a fim de se obter estimativas precisas de área, de atraso crítico e de potência². Tais estimativas foram usadas, juntamente com resultados de taxa de acertos da classificação, para estabelecer as comparações apresentadas na seção 4.5.

Na seção 4.6 é apresentada uma otimização no esquema OAA, a qual originou o esquema batizado de **OAA-melhorado**.

Os resultados obtidos neste capítulo são utilizados como ponto de partida para a discussão dos classificadores baseados em histogramas 2-D apresentados no capítulo 5.

¹ A etapa de classificação corresponde à etapa de teste, na nomenclatura usada em *machine learning*.

² A precisão advém do uso de ferramenta de EDA e biblioteca de células comerciais e também do fato da síntese em si utilizar apenas as portas lógicas necessárias, ao contrário da síntese para FPGA, que precisa fazer uso de elementos configuráveis, os quais possuem uma granularidade maior do que as portas disponíveis em uma biblioteca *standard cell*. Entretanto, posteriormente constatou-se que as plataformas de hardware adotadas em *Software-Defined Radio* (SDR), principal candidato ao uso de *Automatic Modulation Classification* (AMC), baseiam-se em FPGA sobretudo devido à possibilidade de reconfiguração (HARUYAMA, 1999),(DICK; PEDERSEN, 2001),(HARADA, 2007),(MINDEN et al., 2007).

4.1 ANÁLISES NUMÉRICAS PARA DEFINIÇÃO DE ARQUITETURAS DEDICADAS PARA CLASSIFICADORES SVM-HISTO 1-D

Uma vez que o objetivo principal deste trabalho é identificar, projetar, descrever e sintetizar arquiteturas de hardware para classificação de modulações digitais baseadas em SVMs que possam ser implementados em hardware de forma eficiente, as avaliações realizadas nessa seção utilizando *scripts* Matlab[®] levam em conta o tamanho da arquitetura resultante. Assim, no momento de gerar os *scripts* para avaliação da taxa de acertos do classificador, optou-se por avaliar métodos de decisão para múltiplas classes com potencial para gerar hardware com a maior eficiência energética e desempenho possíveis. Neste sentido, um dos pontos que mais causa impacto na quantidade de recursos necessários para a implementação do classificador em hardware é o método utilizado para resolver o problema de classificação de múltiplas classes, conforme já apresentado na seção 2.3.4. Neste capítulo são explorados os aspectos práticos (como tamanho da arquitetura e taxa de acertos da classificação, por exemplo) ao se adotar o método OAO ou o método OAA³.

A avaliação das arquiteturas de classificadores fez uso de duas ferramentas de software: o Matlab[®] e o *Waikato Environment for Knowledge Analysis* (WEKA) (HALL et al., 2009). O Matlab[®] foi usado para gerar sequências de amostras randômicas⁴, segundo as modulações PAM4, BPSK, PSK8 e QAM16, bem como para outras tarefas associadas às características dos sinais (normalização de energia da constelação de símbolos associada a cada modulação, adição do ruído Gaussiano randômico, etc...). O modelo Matlab[®] empregado foi adaptado daquele utilizado pelo trabalho de Cardoso, Castro e Klautau (2013), o qual assume que as amostras complexas do sinal são representadas na forma polar (módulo e fase). Já o WEKA foi utilizado na fase de treinamento das SVMs.

A sequência de operações dos *scripts* Matlab[®] pode ser resumida como segue:

1. Geração randômica de sinais modulados (constelação de símbolos) para treino das SVMs. Para as análises realizadas neste capítulo foram utilizadas as modulações PAM4, BPSK, PSK8 e QAM16;
2. Adição de ruído Gaussiano aos sinais gerados no passo 1 (para as aná-

³Da literatura sabe-se que estes são os métodos mais promissores para implementação em hardware devido ao fato de apresentarem menor custo computacional (LORENA; CARVALHO, 2008), (HSU; LIN, 2002).

⁴Este trabalho de mestrado busca determinar qual modulação é utilizada em uma determinada transmissão e não seu protocolo de transmissão. Por esse motivo, a sequência de amostras é randômica e não contém campos pré-definidos como o preâmbulo, por exemplo.

lises realizadas neste capítulo, a faixa utilizada é de 0 dB a 15 dB, com passo de 1 dB);

3. Extração dos histogramas de módulo e fase dos sinais preparados nos passos 1 e 2;
4. Treino das SVMs com os histogramas extraídos no passo anterior;
5. Geração randômica de novos sinais modulados para realizar a etapa de classificação;
6. Adição de ruído Gaussiano aos sinais preparados no passo 5 (para as análises realizadas neste capítulo, a faixa utilizada é de 0 dB a 15 dB, com passo de 1 dB);
7. Extração dos histogramas de módulo e fase dos sinais preparados nos passos 5 e 6;
8. Classificação dos sinais preparados nos passos 5, 6 e 7 com as SVMs treinadas de acordo com os passos 1, 2, 3 e 4.

Na etapa de treino, para cada modulação a ser avaliada (PAM4, BPSK, PSK8 e QAM16), 300 conjuntos contendo 250 amostras randômicas cada foram gerados ⁵. Para realizar tal tarefa, foram utilizadas funções Matlab[®] para gerar vetores com valores aleatórios de bits (*bit stream*), bem como a constelação para cada modulação a ser avaliada. Esses vetores de bits são então modulados utilizando a constelação gerada para cada modulação, produzindo uma sequência de símbolos que corresponde ao sinal modulado. A esse sinal é então adicionado o ruído Gaussiano, o qual varia na faixa de 0 dB até 15 dB em passos de 1 dB. Na sequência, o *script* Matlab[®] calcula os histogramas de módulo e fase do sinal gerando um arquivo que é então utilizado para realizar o treino das SVMs. O mesmo procedimento foi realizado para gerar a sequência de símbolos a ser utilizada na etapa de classificação, com a diferença que somente 50 conjuntos de 250 amostras são necessários para realizar a classificação ⁵.

Após a etapa de geração das sequências de símbolos para treino e classificação (teste), os arquivos obtidos foram utilizados com o software WEKA para realizar o treino e a classificação das amostras geradas. Sucessivas chamadas do software WEKA foram realizadas a partir do *script* Matlab[®] passando, entre outros parâmetros, o nome do arquivo com as amostras de treino,

⁵Para esta etapa do trabalho adotaram-se os tamanhos dos conjuntos de treino e classificação (teste) estabelecidos por Cardoso, Castro e Klautau (2013), uma vez que, devido a um acordo de cooperação com aquele grupo, teve-se acesso a uma infraestrutura básica para simulação e testes.

o nome do arquivo com as amostras para classificação e o esquema de decisão para resolver o problema de múltiplas classes. O nome do arquivo de saída que recebe o resultado da classificação e os coeficientes dos vetores de suporte, também foi passado para o WEKA. Na etapa de treino das SVMs⁶, o software WEKA forneceu como saída um arquivo contendo os coeficientes dos vetores de suporte a serem utilizados para gerar uma memória de coeficientes de módulo e uma memória de coeficientes de fase para posterior experimentação com a arquitetura sintetizada (ou seja, os valores w_j da equação 2.11).

De acordo com a seção 2.3.3, é possível fazer uma implementação eficiente da equação 2.11 sem multiplicadores e sem ter que calcular o histograma do sinal. Na prática, cada nova amostra de sinal recebida pode ser correlacionada com dois coeficientes: um armazenado na memória de coeficientes de módulo e outro armazenado na memória de coeficientes de fase⁷. O trabalho computacional se resume então em somar e acumular os valores de w_j da equação 2.11.

Para avaliar a taxa de acertos da classificação, as arquiteturas desenvolvidas para cada um dos métodos foram descritas em *Verilog* e sintetizadas com a ferramenta *Synopsys® Design Compiler* (SYNOPSYS, 2011) para uma biblioteca *standard cell* comercial de 90 nm (SORATO et al., 2013b) e, em um passo final, a arquitetura desenvolvida para cada um dos métodos foi simulada com a ferramenta *Modelsim* (Mentor Graphics, 2011).

O resultado da classificação é comparado com o resultado da classificação feita pelo software *WEKA*, o qual também é utilizado como *Golden Model* nesta etapa.

4.2 DATAPATH DE UM BLOCO SVM LINEAR USANDO HISTOGRAMA 1-D

Levando em consideração que o processo de classificação de modulação digital tende a tornar-se uma tarefa cada vez mais comum em dispositivos de comunicação, a arquitetura deve consumir a menor quantidade de energia possível sem comprometer o desempenho, inclusive considerando seu uso em dispositivos móveis. Desta forma, o *datapath* da SVM⁸ foi projetado tão simples quanto possível. A Figura 11 apresenta o *datapath* da SVM linear baseada no HISTO-1D, e que possui os recursos necessários para realizar os

⁶Para cada método de decisão para classificação de múltiplas classes, uma quantidade diferente de SVMs é necessária (Seção 2.3.4).

⁷A amostra do sinal é representada por um número complexo na forma polar.

⁸Neste trabalho considera-se a SVM binária.

cálculos da equação 2.11: dois blocos de memória ROM⁹, um para armazenar os coeficientes do módulo e outro para armazenar os coeficientes da fase do sinal, dois somadores e um registrador.

Inicialmente, seguindo a proposta de Cardoso, Castro e Klautau (2013), assumiram-se 32 *bins* para o módulo e 32 *bins* para a fase do sinal. Portanto, cada bloco de memória deve armazenar 32 coeficientes, gerados na etapa de treino, e assim são necessários 5 bits ($n = m = 5$ na Figura 11) de endereçamento para cada bloco de memória. Os *scripts* Matlab[®] levam em consideração o máximo tamanho dos coeficientes gerados pelo software WEKA, de forma que é possível armazenar cada coeficiente utilizando 16 bits. Desta forma, cada linha da memória terá um tamanho de 16 bits ($k = 16$ na Figura 11).

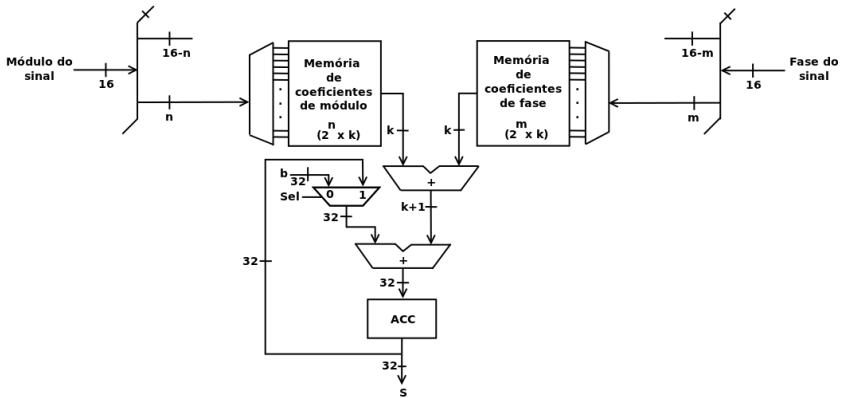


Figura 11 – *Datapath* de um bloco SVM

O comportamento do *datapath* é como segue: quando a primeira amostra complexa do sinal chega ao classificador, os coeficientes de módulo e fase relacionados são lidos das memórias e somados usando o primeiro somador. O resultado de tal adição é somado ao valor b^{10} (apresentado na Equação 2.11) usando o segundo somador e seu resultado é armazenado no registrador ACC. Após, para cada nova amostra que chega, a operação se repete, com a diferença de que o segundo somador é usado para calcular o total acumulado (sinal *sel* = 1 na Figura 11), o qual é escrito no registrador ACC.

⁹Embora o uso de memórias RAM permitiria maior flexibilidade à arquitetura, seria necessário estimar um tamanho máximo para os blocos de memória. Assim, decidiu-se usar blocos de memórias ROM, cada um com o tamanho exato para armazenar os coeficientes necessários a cada versão de *datapath* investigado.

¹⁰A análise dos parâmetros apresentados pelo software WEKA mostrou que este valor precisa ser representado com 32 bits.

Para cada nova amostra complexa que chega, os **n** bits mais significativos do módulo são utilizados para selecionar um coeficiente na memória ROM do módulo e, simultaneamente, os **m** bits mais significativos da fase são utilizados para selecionar um coeficiente na memória ROM da fase. A operação realizada é equivalente à soma ponderada dos *bins* dos histogramas de módulo e fase do sinal complexo recebido.

Para ambos métodos de classificação adotados (descritos nas próximas duas seções), esse processo é repetido até que 50 amostras¹¹ sejam recebidas. Assim que a última amostra é processada, a classificação é realizada. É importante notar que, conforme mencionado na subseção 2.3.3, a escolha do extrator de parâmetros **HISTO-1D** permite a construção de um *datapath* sem multiplicadores, o que contribui para construir um hardware mais simples, ocupando menor área. Entretanto, vale observar que isto somente é possível porque as amostras complexas do sinal vão sendo processadas à medida que vão chegando.

As seções 4.3 e 4.4 apresentam, respectivamente, uma arquitetura de hardware para classificador baseado no esquema OAO e uma arquitetura de hardware para classificador que segue o esquema OAA. Ambas arquiteturas utilizam o *datapath* da Figura 11 como elemento básico.

4.3 ARQUITETURAS SVM-HISTO COM ESQUEMA DE DECISÃO *ONE AGAINST ONE* (OAO)

Conforme apresentado na Equação 2.3.4, o método OAO requer um número de SVMs que cresce como uma combinação dois a dois do número de classes a classificar. A Figura 12 apresenta o diagrama de blocos de um classificador para quatro modulações (conforme considerado no presente capítulo) que utiliza este método.

O bloco **Extrator de características** entrega as amostras complexas do sinal em banda base, conforme apresentado na seção 2.1. Cada bloco SVM é treinado para identificar uma entre duas classes (modulações).

O bloco **Votador**, detalhado na Figura 13, é um discriminador que identifica qual das classes recebeu o maior número de “votos”¹² e funciona como segue. Cada bloco SVM entrega um número que corresponde à soma ponderada dos *bins* do histograma do sinal na entrada. A discriminação entre cada classe ocorre através do bit de sinal do valor de saída (representado no bit

¹¹Nesta etapa do trabalho, seguiu-se o tamanho do conjunto de classificação (testes) ótimo conforme determinado por Cardoso, Castro e Klautau (2013).

¹²Na prática, esse bloco determina qual das classes apresenta o maior valor (i.e., a maior soma ponderada).

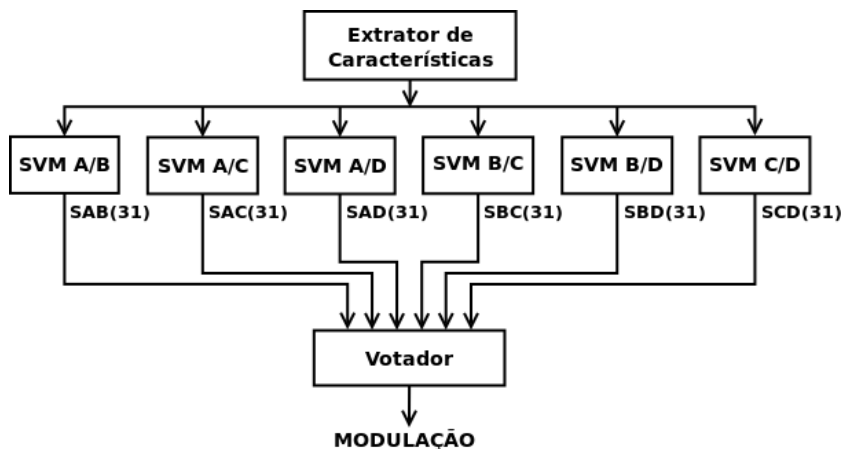


Figura 12 – Diagrama em blocos do classificador baseado no esquema *one against one*.

31). Por exemplo, **SAB**, **SAC** e **SAD** serão valores positivos (bit 31 igual a 1) se a classe de entrada for **A**. Porém, se a classe de entrada for **B**, **C** ou **D**, **SAB**, **SAC** ou **SAD** será negativo, respectivamente. Desta forma, a saída de cada SVM é combinada através de seus bits de sinal correspondentes, por meio de somadores, como mostrado na Figura 13. A tarefa do bloco **Comparador** é determinar qual das entradas possui o maior valor, determinando assim qual a entrada candidata a representar a classe (modulação) correta.

O bloco **Comparador** tem um registrador de dois bits para cada uma das entradas para armazenar a saída de cada SVM. A carga destes registradores ocorre mediante um sinal de controle que é gerado após todas as amostras para classificação terem sido processadas pelo *datapath* apresentado na seção 4.2¹³.

¹³Os sinais de controle foram omitidos nas Figuras 11 e 12.

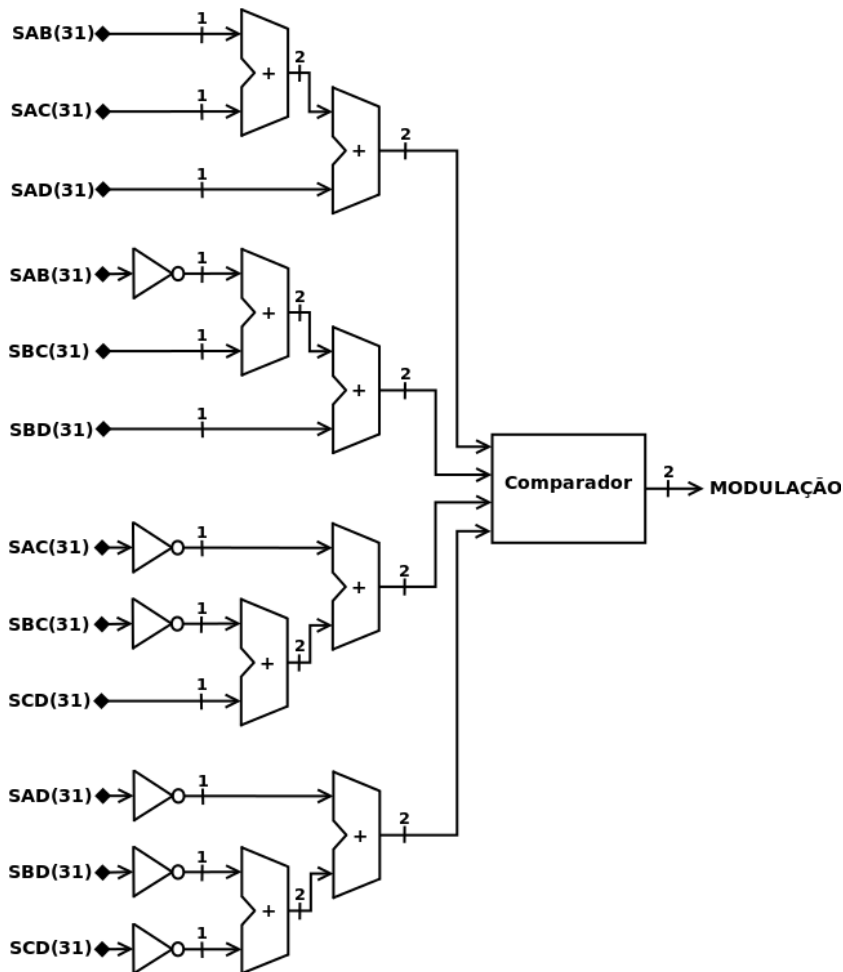


Figura 13 – Hardware do bloco **Votador**.

4.4 ARQUITETURAS SVM-HISTO COM ESQUEMA *ONE AGAINST ALL* (OAA)

Conforme apresentado na seção 2.3.4, para o método OAA o número de SVMs necessárias é igual ao número de classes a serem classificadas. O diagrama em blocos para este esquema considerando quatro modulações é apresentado na Figura 14.

Neste método, cada SVM é treinada para identificar uma dada modulação entre todas a serem classificadas. A saída de cada bloco SVM é um número positivo caso a entrada seja a classe para a qual a SVM tenha sido treinada e negativo caso contrário.

Desta forma, para identificar a classe correta, basta identificar qual das saídas é positiva, o que pode ser realizado com um simples comparador, tarefa realizada pelo bloco **Verificador** na Figura 14.

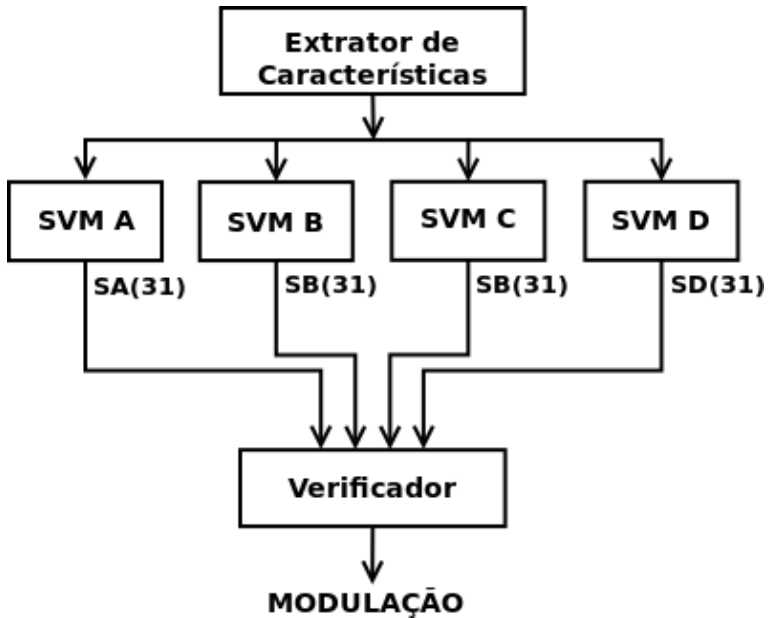


Figura 14 – Diagrama em blocos do esquema *one against All*.

Da mesma forma que no caso do método OAO, o bloco **Verificador** somente faz a classificação final após todas as amostras para classificação terem sido processadas pelo *datapath* apresentado na seção 4.2. Um sinal de controle é utilizado para indicar que a classificação foi completada e a saída do classificador apresenta um resultado estável¹⁴.

A próxima seção apresenta resultados de síntese das arquiteturas completas dos classificadores baseados no esquema de decisão de classificação de múltiplas classes OAO e OAA. São realizadas avaliações de taxa de acertos da classificação, área, consumo de energia e desempenho (frequência máxima da arquitetura).

¹⁴Os sinais de controle foram omitidos nas Figuras 13 e 14.

4.5 COMPARAÇÃO DOS RESULTADOS DA SÍNTESE DE ARQUITETURAS SVM-HISTO UTILIZANDO ESQUEMA OAO COM ARQUITETURAS SVM-HISTO UTILIZANDO ESQUEMA OAA

Esta seção apresenta uma avaliação das arquiteturas baseadas em OAO e OAA, com o objetivo de definir a melhor opção de classificador quando são levados em conta taxa de acertos da classificação, área ocupada pelo hardware, frequência máxima de operação, potência e energia por classificação.

Conforme já mencionado na Seção 4.2, para esta avaliação considerou-se o uso de 32 *bins* ($n = m = 5$ na Figura 11).

As arquiteturas foram descritas em *verilog* e sintetizadas para uma biblioteca *standard cell* comercial de 90 nm com a ferramenta *Design Compiler* (SYNOPSIS, 2011). A Tabela 2 apresenta os resultados de síntese para as arquiteturas baseadas em OAO e OAA.

	OAO	OAA
Tamanho dos <i>Bins</i> (módulo/fase)	32/32	32/32
Área (μm^2)	14743	9921
Atraso crítico(ns)	3,32	3,31
Potência (μW)	150	102
Energia por classificação (nJ)	4,63	3,15

Tabela 2 – Resultados da síntese lógica OAO X OAA.

Uma das primeiras observações que se pode fazer da Tabela 2 é que o atraso crítico é praticamente o mesmo para ambos os casos. Isso se deve ao fato de que o mesmo *datapath* é utilizado, diferindo apenas no esquema de decisão para classificação de múltiplas classes, o qual não aumenta significativamente o caminho crítico. A arquitetura que utiliza o esquema OAA é menor e consome menos energia porque utiliza menos SVMs, conforme discutido na Seção 2.3.4. Na tabela, pode-se observar diferenças de 32% na potência consumida e de 32,71% na área total, em favor da arquitetura baseada em OAA.

Durante os testes com OAO e OAA percebeu-se que muitos *bins* do módulo não carregam informação alguma. Essa observação leva a uma possibilidade de otimização na arquitetura, uma vez que estes valores podem ser descartados, diminuindo o tamanho da memória necessária para armazenar os coeficientes de módulo. Análises numéricas com as modulações adotadas para esta etapa da avaliação (BPSK, PAM4, QAM16 e PSK8) mostraram

que é possível reduzir a quantidade de *bins* de módulo dos 32 *bins* assumidos originalmente para 16 *bins*, sem redução significativa na taxa de acertos do classificador.

Desta forma, o valor **n** (endereçador da memória de coeficientes de módulo), na Figura 11 pode ser reduzido para 4 (16 *bins*). Note que reduzir o número de *bins* significa aumentar a largura dos mesmos, uma vez que a faixa de contagem continua a mesma. Novas arquiteturas foram descritas para esta nova situação e os resultados comparativos para todas essas situações são apresentados na Tabela 3. Nesta tabela são apresentados os resultados da síntese (*standard cells* de 90 nm) para os quatro casos de OAO e OAA com 32 e 16 *bins* para o módulo.

Tamanho dos <i>Bins</i> (módulo/fase)	OA0		OAA	
	32/32	16/32	32/32	16/32
Área (μm^2)	14743	13952	9921	9378
Atraso crítico(ns)	3,32	3,28	3,31	3,29
Potência (μW)	150	143	102	98
Energia por classificação (nJ)	4,63	4,41	3,15	3,02

Tabela 3 – Comparação da síntese lógica OAO X OAA, 16 X 32 *bins* para módulo.

Como era de se esperar, o atraso crítico continua sendo aproximadamente o mesmo para todas as arquiteturas.

Comparando as arquiteturas com 16 *bins* de módulo e 32 *bins* de fase (16/32) percebe-se uma redução de 31,46% na potência e de 32,78% na área total, em favor da arquitetura baseada em OAA. Como pode ser observado, a proporção de redução de potência e área permanecem basicamente as mesmas. No entanto, quando se compara as arquiteturas OAO 16/32 e 32/32 percebe-se uma redução de 4,6% para a potência e de 5,36% para a área total, em favor das arquiteturas 16/32. Da mesma forma, pode-se observar uma redução de 5,5% na potência e de 4,03% na área total das arquiteturas baseadas em OAA 16/32, em relação às arquiteturas OAA 32/32. Outra comparação possível de realizar entre as arquiteturas OAA e OAO é a energia por classificação para arquiteturas 32/32 e arquiteturas 16/32, a qual apresenta uma redução de, respectivamente, 31,96% e 31,52% em favor de arquiteturas baseadas em OAA.

Uma vez que o atraso crítico é aproximadamente o mesmo para todos os casos e que a arquitetura OAA ocupa menos área e consome menos energia, o esquema de decisão para classificação de múltiplas classes OAA é a opção mais indicada para implementação do classificador SVM-HISTO-1D

em hardware dedicado.

Apesar de as arquiteturas com 16 *bins* de módulo ocuparem menos área e demandarem uma potência menor, ainda é necessário avaliar a taxa de acertos da classificação quando se reduz o número de *bins* para representar o módulo. O gráfico da Figura 15 apresenta o resultado de classificação para as quatro arquiteturas discutidas. Estes resultados foram obtidos simulando-se as arquiteturas com a ferramenta ModelSim (Mentor Graphics, 2011) e comparando-se os resultados de classificação com aqueles esperados para um conjunto de sinais de entrada gerados pelo *script* Matlab[®] (passos 4 e 5 do fluxo apresentado na seção 4.1).

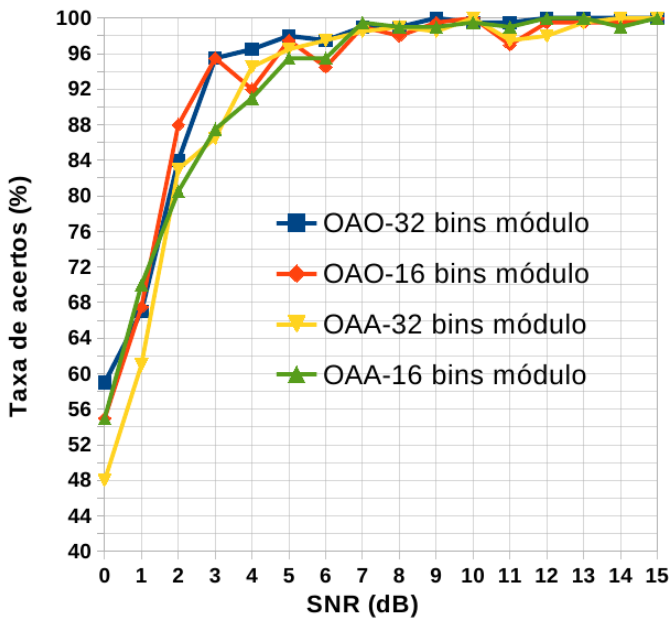


Figura 15 – Taxa de acertos dos classificadores baseados em *OAO* e *OAA*.

No gráfico da Figura 15 pode-se observar que, para valores abaixo de 4 dB, o método OAO apresenta resultados de classificação ligeiramente melhores do que o método OAA. Entretanto, de maneira geral, todas as curvas apresentam resultados semelhantes para valores de *Signal-to-Noise Ratio* (SNR) acima de 3 dB. Então, considerando que a menor taxa de acertos do método OAA é suplantado menor área e menor energia por classificação, conclui-se que o método OAA é um bom candidato para implementação de

classificadores de múltiplas classes baseados em SVM.

Com relação à redução do número de *bins* do módulo, quando se comparam as curvas OAA-32 *bins* e OAA-16 *bins*, percebe-se que existe uma pequena redução na taxa de acertos da classificação da arquitetura para valores de SNR entre 3 dB e 7 dB. No entanto, comparando essas mesmas curvas para 0 dB, a OAA-16 apresenta uma taxa de acertos aproximadamente 7% melhor. Este fato provavelmente ocorre porque quando o número de *bins* diminui, a largura do mesmo aumenta, o que acaba resultando em uma maior imunidade ao ruído (SNR) (fato observado entre 0 dB e 2 dB).

Uma vez que a arquitetura baseada em OAA apresenta vantagens em termos energéticos (menor potência e menor energia por classificação) e em termos de área, e portanto é a mais indicada para a tarefa de classificação de modulação digital, é importante investigar os motivos pelos quais o resultado de classificação desta é pior do que o da arquitetura baseada em OAO. A próxima seção apresenta tal análise, juntamente com uma otimização do classificador baseado em OAO.

4.6 OTIMIZAÇÃO DA ARQUITETURA BASEADA NO ESQUEMA OAA

Durante os experimentos com a arquitetura baseada no esquema OAA observou-se que parte dos erros de classificação apresentados se devem a “empates” das saídas das SVMs. Isto é, mais do que uma SVM identificou a amostra de entrada como sendo pertencente à classe para a qual foi treinada.

O esquema OAA original prevê que a classificação é realizada treinando uma SVM para identificar uma determinada classe dentre as demais. Se a entrada da SVM contiver amostras de uma classe para a qual a SVM foi treinada, a saída será positiva, caso contrário, será negativa. Porém, quando a SNR diminui, podem ocorrer casos em que mais de uma saída seja positiva ou mesmo, que todas sejam negativas. A solução dos empates deve acarretar em um melhor desempenho na taxa de acertos do classificador.

Para resolver o problema, substituiu-se o bloco **Verificador** da Figura 14 por um bloco mais complexo, o **Verificador melhorado**, que basicamente faz a comparação das magnitudes das saídas das SVMs e determina o maior valor dentre estas. Este esquema foi batizado de **OAA melhorado**. A Figura 16 apresenta o bloco **Verificador melhorado** proposto.

Basicamente, o bloco **Verificador melhorado** é composto de três comparadores de magnitude (considerando nosso exemplo de 4 classes) que determinam qual das saídas apresenta o maior valor.

A Figura 17 apresenta um gráfico comparativo da taxa de acertos da classificação dos três métodos: OAO, OAA e OAA melhorado. No gráfico,

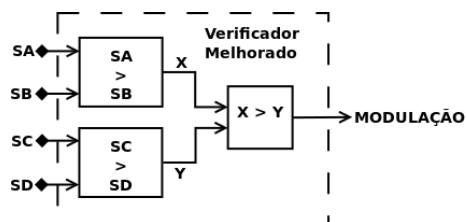


Figura 16 – Diagrama em blocos do **Verificador melhorado**.

é possível verificar que o esquema OAA melhorado apresenta resultados de classificação melhores, principalmente para valores abaixo de 7 dB. Em especial, para 0 dB, o esquema OAA melhorado com 32 *bins* apresenta taxa de acerto 26% maior do que a do esquema OAA com 32 *bins*.

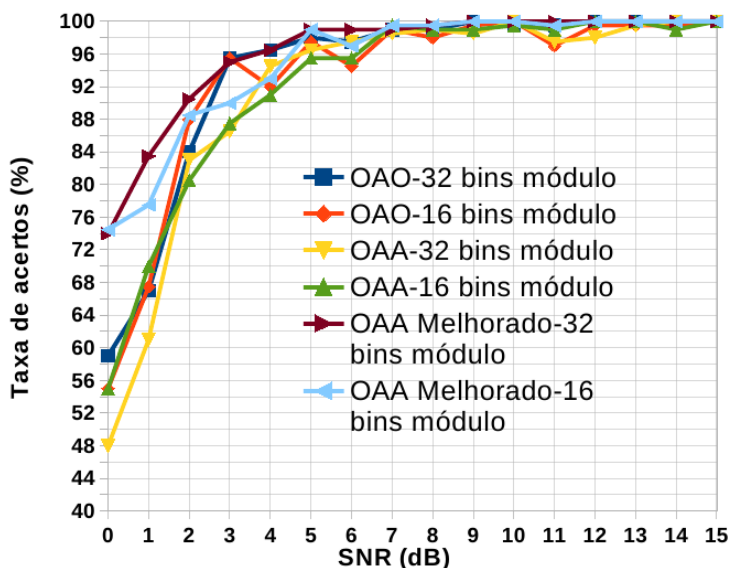


Figura 17 – Taxa de acertos dos classificadores baseados em OAO, OAA e OAA melhorado

Estes resultados mostram que o esquema OAA melhorado resolve o problema de baixa taxa de acertos apresentado pelo esquema OAA original e

portanto, constitui-se em uma boa solução para o problema de classificação de múltiplas classes. Contudo, pode ser observado na Figura 17 que o esquema OAA melhorado com 16 *bins* apresenta maior variação na taxa de acertos da classificação do que o OAA melhorado com 32 *bins*. Devido a esse fato, foi tomada a decisão de projeto de abandonar a síntese para 16 *bins*.

A arquitetura OAA melhorado foi descrita em *verilog* e sintetizada para uma biblioteca *standard cell* comercial de 90 nm com a ferramenta *Design Compiler*. A Tabela 4 apresenta os resultados da síntese.

	OA0	OA	OA melhorado	OA0xOA melhorado (%)	OA0xOA melhorado (%)
Tamanho dos <i>Bins</i> (módulo/fase)	32/32	32/32	32/32	32/32	32/32
Área (μm^2)	13952	9378	11987	-14,08	+27,82
Atraso crítico (ns)	3,28	3,29	3,15	-3,96	-4,25
Potência (μW)	143	98	100	-30,07	+2,04
Energia por classificação (nJ)	4,41	3,02	3,08	-30,16	+1,98

Tabela 4 – Comparação de resultados da síntese lógica para as arquiteturas baseadas em OA0, OA e OA *melhorado*.

Para aproximar os testes realizados de uma situação prática, a frequência de *clock* foi determinada a partir de um sinal modulado em QAM16 em uma taxa de dados de 54 *Mbps*, que representa um sinal comumente encontrado em comunicações digitais. Assim, a frequência mínima para processar esse sinal é 16 *MHz* (período de 61,3ns), bem abaixo da frequência máxima que as arquiteturas projetadas podem atingir¹⁵.

A Tabela 4 mostra o resultado das sínteses das arquiteturas OA0, OA e OA melhorado assumindo-se 16 *MHz* como frequência de relógio. Pode-se observar que o OA melhorado apresenta uma área 14,08% menor do que o OA0 e 27,82% maior do que o OA. Desta comparação pode-se concluir que o esquema OA melhorado ocupa uma área maior do que o OA original devido ao bloco **Verificador melhorado**. Apesar da diferença de área, todos os três classificadores apresentam aproximadamente o mesmo atraso crítico e quase o mesmo número de ciclos de *clock* para realizar uma classificação, o que significa que o tempo de classificação é aproximadamente o mesmo para os três casos. Apesar de ocuparem áreas significativamente diferentes,

¹⁵Ver estimativas de atraso crítico nas Tabelas 2 e 3.

os esquemas OAA e OAA melhorado apresentam consumos de energia semelhantes, os quais são 31,46% e 30,07% menores do que o do esquema OAO. Estes valores se devem ao fato de que os esquemas OAA precisam de um menor número de SVMs para realizar a classificação. Da mesma forma, a energia por classificação é praticamente a mesma para os esquemas OAA e OAA melhorado.

4.7 CONCLUSÕES DO CAPÍTULO

Nesse capítulo foi observado que a variação da taxa de acertos da classificação da SVM (número de *bins*) tem um impacto limitado nas características de área, potência e energia por classificação.

As análises de taxas de acertos, área, potência e energia por classificação realizadas no início deste capítulo permitiram concluir-se que o esquema de decisão para classificação de múltiplas classes OAA é o mais adequado para implementação de classificadores de modulações digitais em hardware. Porém, para melhorar a taxa de acertos deste esquema, tornando-a equivalente àquela apresentada pelo esquema OAO, foi realizada uma modificação arquitetural no classificador OAA, a qual consistiu no uso de comparadores de magnitude no bloco **Verificador**. Assim, o classificador resultante, baseado no esquema OAA melhorado, apresenta características de área, potência e energia por classificação equivalentes às do esquema OAA, e taxa de acertos equivalente à do esquema OAO.

A fim de se determinar com maior exatidão os limites de uso de classificadores baseados em SVM e histogramas, decidiu-se ampliar e aprofundar as análises, fazendo uso de uma modelagem matemática mais adequada do que a utilizada em trabalhos anteriores (CARDOSO; CASTRO; KLAUTAU, 2013), (SORATO et al., 2013b) e (SORATO et al., 2013a). Em especial, avaliou-se a robustez da técnica em relação a diversas características, assumindo-se que o receptor não conhece a taxa de transmissão. Estas análises, juntamente com a proposta de uso de histogramas 2-D, são apresentadas no capítulo 5.

5 USO DE CLASSIFICADORES BASEADOS EM SVM E HISTOGRAMAS 2-D EM UM AMBIENTE NÃO COOPERATIVO

Este capítulo apresenta uma análise aprofundada da classificação de modulações digitais usando *Support Vector Machine* (SVM) e histogramas 2-D. Esta análise se diferencia daquela descrita no capítulo 4 não apenas pelo uso de histogramas 2-D, mas sobretudo pelo fato de assumir que o receptor não conhece a taxa na qual os dados são transmitidos, o que foi modelado variando-se a taxa na qual o sinal de entrada é amostrado. Tal assertiva, além de diferenciar o presente trabalho de grande parte dos trabalhos correlatos, está mais próxima das situações de uso dos rádios cognitivos. Outras diferenças em relação à análise do capítulo 4 dizem respeito à infraestrutura utilizada, às modulações digitais usadas como estudo de caso (PSK4, PAM4, QAM4, PSK8, QAM16 e QAM64, ao invés de PAM4, BPSK, PSK8 e QAM16) e ao número maior de parâmetros analisados (tamanho do conjunto de treino, *Signal-to-Noise Ratio* (SNR), número de amostras em cada histograma e número de *bins*). Além disso, a faixa de SNR considerada também é maior, o que também aproxima mais a análise realizada às situações encontradas na prática. A análise sistemática e minuciosa da taxa de acertos da classificação em relação a todos estes parâmetros demonstrou o alto grau de robustez do método de classificação baseado em SVM e histogramas. Finalmente, vale observar que as arquiteturas de hardware dedicadas definidas e avaliadas no capítulo 4 também podem ser utilizadas para SVMs com histograma 2-D, necessitando apenas pequenas adaptações, conforme é apresentado na seção 5.5.

Baseando-se nos resultados das análises realizadas, são apresentadas as modificações necessárias na arquitetura do classificador baseado em histogramas 1-D do capítulo 4 para viabilizar a classificação com histogramas 2-D, levando em conta que o transmissor e o receptor não estão sincronizados. A seguir, são apresentadas comparações de desempenho e uso de recursos de hardware para os classificadores baseados em *Histogramas 1-D* (HISTO-1D) e em *Histogramas 2-D* (HISTO-2D) propostos, quando sintetizados para *Field-Programmable Gate Arrays* (FPGAs) da Altera. A opção por FPGAs como forma de implementação deve-se ao fato de a maioria das plataformas de hardware para *Software-Defined Radio* (SDR) disponíveis no mercado utilizarem este tipo de dispositivo (conforme já exposto na seção 1.5).

Este capítulo está organizado da seguinte forma. A seção 5.1 apresenta uma análise da modelagem de sinais usando histogramas 1-D e 2-D e aponta as vantagens de se usar histogramas 2-D ao invés de histogramas 1-D para a

classificação de modulações digitais. A seção 5.2 apresenta a infraestrutura utilizada para realizar as avaliações dos classificadores baseados em SVM e HISTO-2D. Os resultados destas avaliações são apresentados na seção 5.3. Parâmetros como tamanho do conjunto de treino, SNR, número de amostras em cada histograma e número de *bins* são avaliados. A seção 5.4 apresenta uma comparação entre as taxas de acertos dos classificadores HISTO-2D e HISTO-1D quando o número de *bins* é variado. A seção 5.5 apresenta a arquitetura resultante para classificadores HISTO-2D, bem como uma comparação dos resultados de síntese para FPGA de classificadores HISTO-1D e HISTO-2D.

5.1 ANÁLISE DA MODELAGEM DE SINAIS USANDO HISTOGRAMAS 1-D E 2-D

O sinal recebido na antena do receptor inicialmente sofre um processamento analógico (filtragem analógica, amplificação, etc) para posteriormente ser convertido para digital por um conversor A/D. Após essa etapa, o sinal complexo é extraído pelo bloco Demodulador IQ (ver Figura 4). Esta sequência de amostras complexas é então utilizada pelo classificador de modulação.

Para uma sequência de N amostras complexas, existem duas formas de se obter o histograma. A forma proposta por Cardoso, Castro e Klautau (2013), batizada de HISTO-1D (e também adotada por Sorato et al. (2013b)) consiste em construir um histograma para o eixo real e outro para o eixo imaginário e posteriormente, concatená-los para usar na classificação baseada em SVM. Nesta seção, propõe-se uma forma alternativa, batizada de HISTO-2D, a qual combina as amostras do eixo real e do eixo imaginário, criando uma "matriz", a qual será usada para a classificação também baseada em SVM. Desta forma, a abordagem 2-D leva em conta a correlação entre a parte real e a parte imaginária do sinal, o que não ocorre no caso da abordagem 1-D.

As Figuras 18 e 19 mostram imagens de histogramas 2-D para sinais nas modulações consideradas nesta seção (PSK4, PAM4, QAM4, PSK8, QAM16 e QAM64). Nestas imagens, as regiões mais escuras significam uma alta contagem no respectivo *bin*, ao passo que branco significa que a contagem é zero. Adotando-se uma resolução de 32 *bins* para cada eixo, tem-se um total de 1024 *bins*.

Os sinais utilizados para gerar esses histogramas foram sobreamostrados com um fator de 80, o qual é equivalente a uma representação do sinal contínuo em ambiente Matlab® sem perdas significativas para o processamento necessário (levando em conta que o sinal está centrado na frequência

intermediária de 70 MHz, conforme exposto na seção 2.1) e foi determinado no livro de Klautau (ca. 2014)¹. Cada histograma apresentado nas Figuras 18 e 19 corresponde a uma sequência de 400 símbolos. Os histogramas apresentados na Figura 18 tem SNR = 10 dB enquanto que os histogramas apresentados na Figura 19 tem SNR = -10 dB. Ao comparar as Figuras 18 e 19 fica clara a influência da SNR no padrão de histogramas a serem classificados pela SVM. Na Figura 18 as imagens são mais nítidas, significando que o padrão está mais definido, ao passo que na Figura 19 o padrão está menos definido. Ao compararmos entre si as imagens para as modulações QAM16 e QAM64 nas Figuras 18 e 19, vemos que é mais difícil distinguir as duas modulações na Figura 19 do que na Figura 18.

As Figuras 20 e 21 mostram imagens similares de histogramas 1-D dos mesmos sinais. Para fins de visualização e comparação, ao invés de apresentar a concatenação dos histogramas 1-D, construíram-se histogramas 2-D utilizando a informação capturada pelo extrator de parâmetros baseado em histogramas 1-D. Para uma comparação mais justa, o número total de *bins* para os dois casos foi mantido igual. Portanto, nas Figuras 20 e 21, o número de *bins* em cada eixo componente é ajustado para 512, o que resulta em um número total de *bins* igual a 1024.

Nas Figuras 18, 19, 20 e 21 pode ser observado que, para uma mesma quantidade de *bins*, o extrator de parâmetros baseado em histogramas 1-D apresenta uma resolução melhor do que o baseado em histogramas 2-D. Contudo, histogramas 1-D não capturam a informação a respeito da correlação das partes real e imaginária do sinal. Por exemplo, dadas duas amostras complexas $a + bi$ e $c + di$, a SVM com o extrator de parâmetros baseado em histogramas 1-D não tem como decidir se as amostras são $a + bi$, $c + di$ ou $a + di$, $c + bi$. Ao desconsiderar a correlação entre as partes real e imaginária perde-se uma parte da informação contida no sinal que é importante para o processo de *Automatic Modulation Classification* (AMC).

As seções que seguem apresentam uma avaliação detalhada do uso de histogramas 2-D no extrator de parâmetros **HISTO**, iniciando pela infraestrutura utilizada para as avaliações numéricas realizadas.

¹O livro ainda não está disponível, porém os artefatos de software estão disponíveis para *download* na página *web* do livro. A utilização destes artefatos de software teve a anuência do autor devido ao acordo de cooperação mútua.

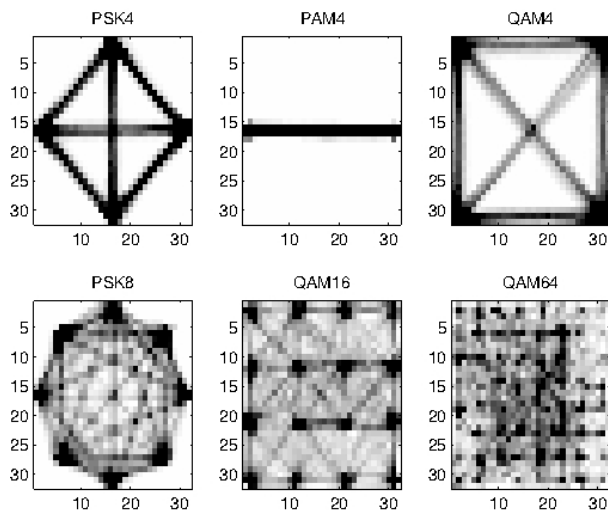


Figura 18 – Histogramas 2-D de sinais com modulações PSK4, PAM4, QAM4, PSK8, QAM16, QAM64 e SNR = 10 dB.

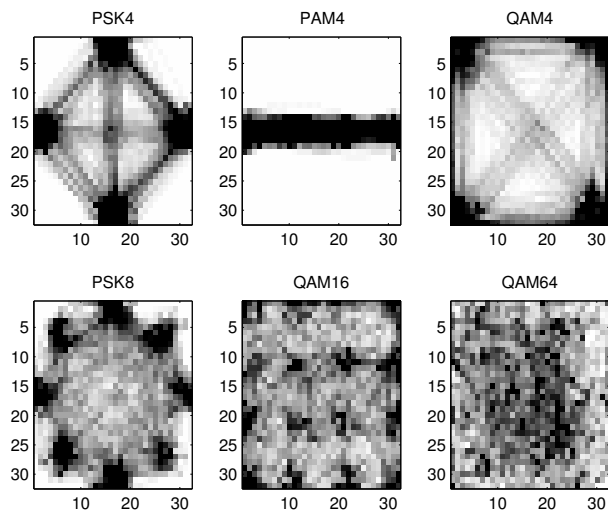


Figura 19 – Histogramas 2-D de sinais com modulações PSK4, PAM4, QAM4, PSK8, QAM16, QAM64 e SNR = -10 dB.

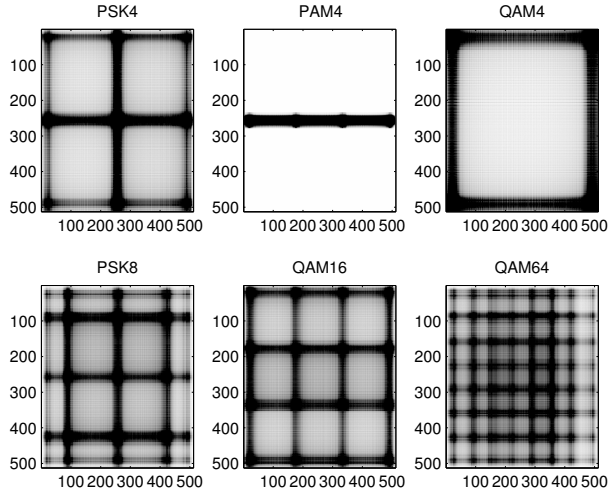


Figura 20 – Construção de histogramas 2-D usando informações 1-D com $\text{SNR} = 10 \text{ dB}$.

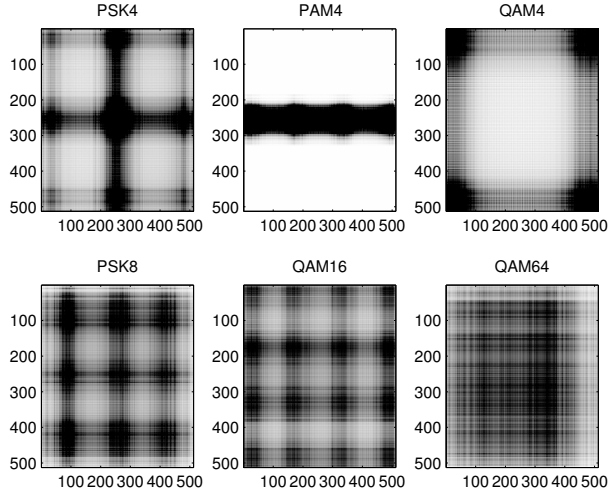


Figura 21 – Construção de histogramas 2-D usando informações 1-D com $\text{SNR} = -10 \text{ dB}$.

5.2 INFRAESTRUTURA PARA ANÁLISE NUMÉRICA DE CLASSIFICADORES BASEADOS EM SVM E HISTOGRAMAS 2-D

Esta seção apresenta a infraestrutura utilizada para realizar as avaliações apresentadas nas demais seções deste capítulo. Semelhantemente ao apresentado no capítulo 4, foram escritos *scripts* em Matlab® para gerar a sequência de dados (*bit stream*), a modulação, a constelação e a sequência de símbolos à qual posteriormente é adicionado o ruído Gaussiano. Essa sequência de símbolos contendo ruído Gaussiano é considerado o sinal recuperado na recepção. O sinal recuperado é amostrado e posteriormente convertido em um sinal complexo por um demodulador de quadratura (bloco Demodulador IQ na Figura 4) que entrega amostras complexas na forma retangular (real e imaginário)². A Figura 22 apresenta um diagrama ilustrativo das etapas processadas pelos *scripts* Matlab® contendo a representação do processo de transmissão (geração de *bit stream*, modulação, etc), efeito do canal (adição de ruído) e a etapa de recepção (amostragem, demodulação IQ, etc).

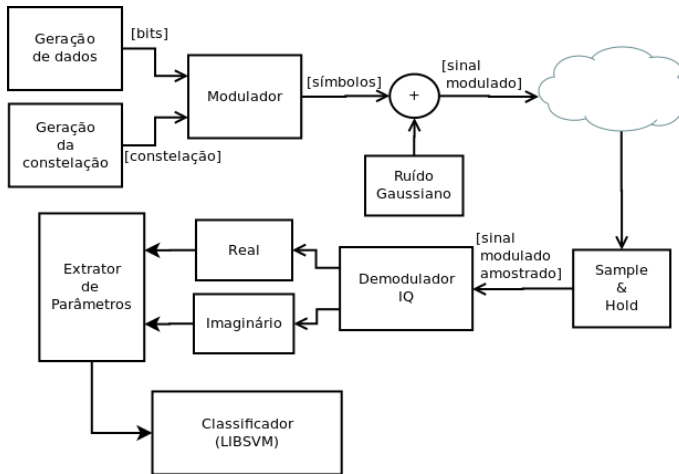


Figura 22 – Diagrama em blocos dos *scripts* Matlab.

Na Figura 22 o bloco "Extrator de Parâmetros" é responsável por construir o histograma a partir das amostras complexas, o qual será utilizado pelo bloco "Classificador (LIBSVM)". Este último bloco é implementado por um

²Aqui reside outra diferença de modelagem em relação à análise realizada no capítulo 4, onde as amostras eram representadas na forma polar (módulo e fase).

software externo chamado **LIBSVM** (CHANG; LIN, 2011)³, o qual substitui o software *Waikato Environment for Knowledge Analysis* (WEKA), utilizado no capítulo 4. A decisão de utilizar o **LIBSVM** ao invés do WEKA deveu-se ao fato do primeiro permitir uma melhor integração com o ambiente Matlab.

O software **LIBSVM** é utilizado para treinar e testar as SVMs de acordo com o parâmetro que se deseja investigar. A partir desta estrutura básica foram escritos *scripts* para testar a robustez ao ruído, a influência do tamanho do conjunto de treino e da granularidade do histograma na taxa de acertos da classificação, bem como para comparação com os resultados apresentados pela solução com histogramas 1-D. Os resultados destas análises são apresentados nas seções seguintes.

5.3 AVALIAÇÃO DA TAXA DE ACERTOS E DA ROBUSTEZ DE AMC BASEADA EM HISTOGRAMAS 2-D

Conforme apresentado na seção 2.2, a etapa de AMC é um passo intermediário entre a detecção do sinal e a demodulação. No contexto de rádio cognitivo e monitoramento espectral "cego", é frequente encontrar ambientes nos quais o transmissor e o receptor não cooperam mutuamente (o que significa que o receptor não tem conhecimento prévio sobre a taxa de transmissão do sinal). Para estes casos, chamados de sistemas de comunicação não-cooperativos, e quando não se faz uma estimativa da taxa de dados e sincronização antes da etapa de AMC, a técnica empregada para obter amostras do sinal é a sobreamostragem do sinal recebido. Portanto, o sistema de AMC deve ser robusto a variações de taxa de transmissão dos dados.

Nesta seção é analisado o impacto de diferentes parâmetros na taxa de acertos e na robustez de AMC baseadas em SVM, principalmente com o objetivo de avaliar o uso de histogramas 2-D. Em relação à análise apresentada no capítulo 4, ampliou-se o número de parâmetros para:

- Tamanho do conjunto de treino (*#training_set*);
- Relação sinal-ruído (SNR);
- Comprimento da sequência (número de amostras em cada histograma) (*seq_len*);
- Número de *bins* (*#nb_bins*).

Além disso, para a análise aprofundada apresentada neste capítulo, também ampliou-se o número de modulações, considerando: PSK4, PAM4,

³Somente durante o desenvolvimento da segunda parte do trabalho descobriu-se essa implementação da SVM.

QAM4, PSK8, QAM16 e QAM64. Para modelar a situação na qual o receptor não conhece a taxa de transmissão, consideram-se diferentes taxas de amostragem do sinal que chega, capturando-se tal informação por meio do parâmetro s_factor . Valores de s_factor maiores que "1" significam sobreamostragem em relação à taxa de transmissão, ao passo que valores menores que "1" significam subamostragem. O sincronismo entre transmissor e receptor é modelado fazendo-se $s_factor = 1$.

Assim, os valores *default* para os parâmetros usados nas avaliações realizadas são:

$$\begin{aligned} \#training_set &= 100 \times 11 \times 6 = 6600 \\ SNR &= \{-40, -35, \dots, 0, 5, 10\} \\ seq_len &= 300 \\ nb_bins &= 8^2 = 64 \\ s_factor &= \{8.0, 5.33, 3.47, 2.28, 1.50, 1.0, 0.66, 0.44, 0.29, 0.19\} \end{aligned}$$

Estes valores constituem uma estimativa inicial para fins de avaliação e, portanto, não são necessariamente os valores ótimos.

Para cada avaliação apresentada nas subseções que seguem, um conjunto de treino e um conjunto de teste (classificação) foram gerados usando os mesmos parâmetros e desta forma, estes conjuntos se diferem pelo ruído Gaussiano, fase e conjunto de bits de informação (*bit stream*) a ser transmitido, os quais foram gerados de forma randômica.

Considerando os valores *default*, cada conjunto de treino/classificação tem 100 histogramas para cada um dos 11 valores de SNR e para cada uma das 6 modulações consideradas, resultando em 6600 histogramas. A SNR é variada uniformemente de -40 dB até $+10$ dB em passo de 5 dB. Cada histograma é extraído de uma sequência de 300 amostras complexas. Cada uma destas amostras é contada uma vez no histograma 2-D (*bin* dado pela correlação entre a parte real e a parte imaginária) e duas vezes no histograma 1-D (um *bin* para parte real e outro *bin* para parte imaginária).

O extrator de parâmetros baseado em histogramas 2-D com 64 *bins* trunca as partes real e imaginária e considera apenas os 3 bits mais significativos, o que corresponde a 8 *bins* para cada eixo. De forma similar, o extrator de parâmetros baseado em histogramas 1-D com 64 *bins* trunca as partes real e imaginária do sinal complexo considerando os 5 bits mais significativos, o que resulta em 32 *bins* para o histograma da parte real e 32 *bins* para o histograma da parte imaginária⁴.

Nas Figuras 23 e 24 são mostradas imagens dos conjuntos de matrizes de treino para histogramas 2-D e para histogramas 1-D, respectivamente.

⁴Nesta seção, quando não for mencionado o contrário, se fará referência ao extrator de parâmetros baseado em histogramas 2-D.

Nestas matrizes, cada linha da ordenada (*training sample*) corresponde a um histograma. Em cada histograma, os pontos mais escuros significam altos valores de contagem. Os histogramas 2-D, originalmente montados em forma de matriz, foram "vetorizados" para caber em uma única linha da ordenada. As matrizes estão organizadas de modo que os histogramas de cada modulação ocupam 100 linhas consecutivas da ordenada, da seguinte forma: PSK4 (linhas 1 a 100), PAM4 (linhas 101 a 200), QAM4 (linhas 201 a 300), PSK8 (linhas 301 a 400), QAM16 (linhas 401 a 500) e QAM64 (linhas 501 a 600). Em ambas Figuras, 23 e 24, é possível ver claramente a distinção entre as 6 modulações.

Para cada matriz de treino existe um vetor de rótulos correspondente que indica a modulação de cada linha. A matriz de treino e o vetor de rótulos são as entradas para a rotina de treino da SVM.

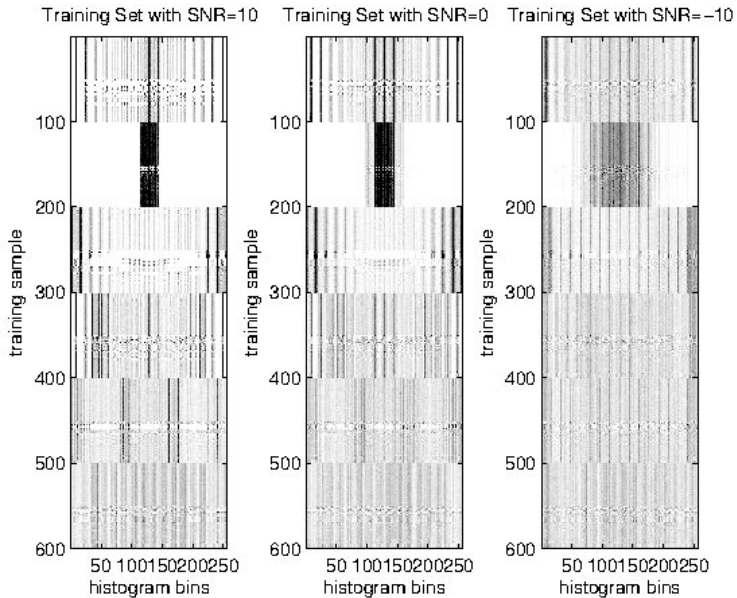


Figura 23 – Matriz representando um exemplo de conjunto de treino com histogramas 2-D. Cada histograma ocupa exatamente uma linha da ordenada (*training sample*), totalizando 100 histogramas para cada uma das seis modulações consideradas: PSK4 (linhas 1 a 100), PAM4 (linhas 101 a 200), QAM4 (linhas 201 a 300), PSK8 (linhas 301 a 400), QAM16 (linhas 401 a 500) e QAM64 (linhas 501 a 600). Em cada histograma, os pontos mais escuros significam altos valores de contagem.

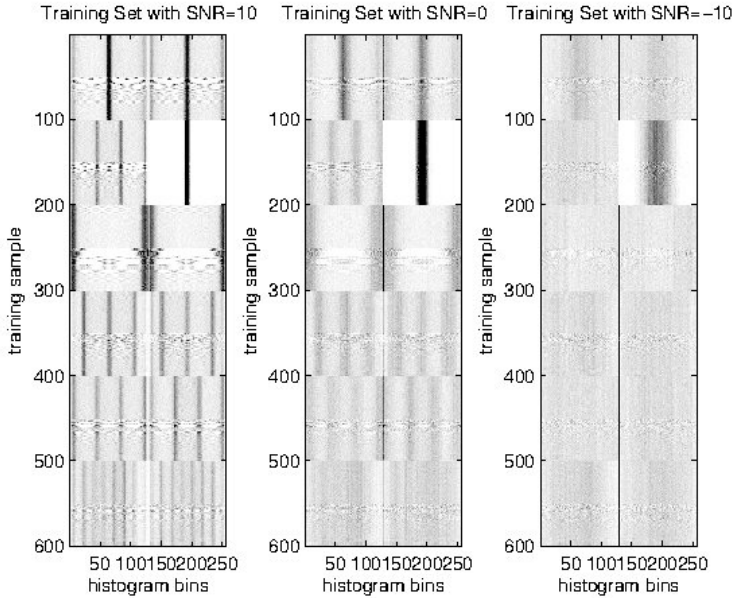


Figura 24 – Matriz representando um exemplo de conjunto de treino com histogramas 1-D. Cada histograma ocupa exatamente uma linha da ordenada (*training sample*), totalizando 100 histogramas para cada uma das seis modulações consideradas: PSK4 (linhas 1 a 100), PAM4 (linhas 101 a 200), QAM4 (linhas 201 a 300), PSK8 (linhas 301 a 400), QAM16 (linhas 401 a 500) e QAM64 (linhas 501 a 600). Em cada histograma, os pontos mais escuros significam altos valores de contagem.

Tanto para o conjunto de treino quanto para o conjunto de classificação, para cada modulação considerada, os sinais foram gerados usando 10 diferentes valores de fator de amostragem. Assim, nas Figuras 23 e 24, para cada 100 histogramas (em 100 linhas consecutivas da ordenada), existem 10 histogramas de cada fator de amostragem (s_factor).

A Figura 25 mostra um zoom do conjunto de treino do histograma 2-D (ver Figura 23) com $SNR = 0$ dB para a modulação QAM4 (linhas 201 a 300), com o objetivo de examinar o efeito de diferentes valores do fator de amostragem na característica dos histogramas. Observa-se que os primeiros 50 histogramas (linhas 201 a 250), os quais correspondem aos valores de amostragem de 8,0, 5,33, 3,47, 2,28 e 1,50 (i.e., sobreamostragem), são bastante uniformes. Porém, com o fator de amostragem igual a 1 (linhas 251

a 260), isto é, o receptor amostrando na taxa de transmissão, os histogramas são menos uniformes. A fase é randômica e isto também causa a variação dentro do grupo de linhas de 251 a 260. Um comportamento similar é observado quando o fator de amostragem é ajustado para 0,66 (linhas 261 até 270). Os 4 últimos valores do fator de amostragem (linhas 261 a 300) são 0,66, 0,44, 0,29 e 0,19, respectivamente, e equivalem a subamostrar o sinal. Nestes casos, se o fator de amostragem é um valor inteiro e múltiplo da taxa de transmissão do sinal, o histograma é passível de variações devido ao efeito randômico da fase.

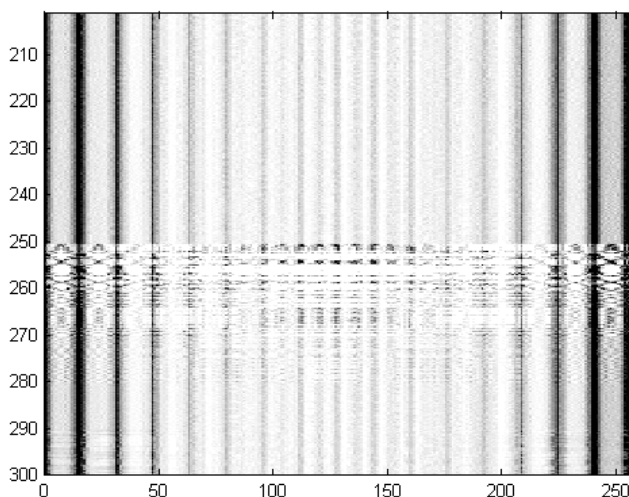


Figura 25 – Conjunto parcial do treino da modulação QAM4.

5.3.1 Tamanho do Conjunto de Treino

Em máquinas de aprendizagem é usual a pergunta *"quantos exemplares são necessários para treinar uma máquina de classificação de forma que tenha um grau de generalidade tal que possa classificar bem novas amostras desconhecidas⁵ ?"*

Esta subseção aborda o caso em que uma SVM é treinada com conjun-

⁵Para o caso em estudo, quantidade de exemplares é a quantidade de histogramas.

tos de treino de tamanhos diferentes e sua taxa de acertos da classificação é avaliada para diferentes valores de SNR. Para tanto, são usados os parâmetros *default* apresentados no início da seção 5.3, exceto pelo tamanho do conjunto de treino, o qual é variado da seguinte forma:

$$\begin{aligned} \#training_set &= \#set_size \times (Num. SNR) \times (Num. Modul.) = \\ &= \{4, 8, 16, 32, 64, 128, 256, 512\} \times \{11\} \times \{6\} \end{aligned}$$

Assim, foram treinados 8 classificadores, cada um com um conjunto de treino de tamanho distinto (pertencente ao conjunto *#set_size*). Cada classificador é composto por 6 SVMs binárias (uma vez que são 6 as modulações digitais consideradas). É importante observar que cada SVM foi treinada para todos valores de SNR de uma vez, ao passo que na fase de classificação, a taxa de acertos foi avaliada separadamente, para cada valor de SNR. Os conjuntos de treino e de classificação foram gerados randomicamente, sendo distintos entre si. A Figura 26 mostra a taxa de acertos da classificação em função da SNR. Para conjuntos de treino pequenos, é observada uma variação aproximadamente uniforme para todos os valores de SNR (ver *#set_size* = 4, 8 e 16). Para valores maiores, as predições das SVMs apresentam menores variações, tornando-se mais estáveis para valores maiores de SNR e de *#set_size*.

Este experimento mostra que adicionar mais exemplares no conjunto de treino faz com que os classificadores SVM apresentem resultados de taxa de acertos da classificação melhores e mais estáveis, tanto para baixos quanto para altos valores de SNR. Esta melhora, no entanto, está limitada a aproximadamente 256 amostras. A partir deste tamanho, aumentar o conjunto de treino não resulta em uma melhora considerável na taxa de acertos da classificação, nem para baixos e nem para altos valores de SNR.

O tamanho do conjunto de treino não tem impacto na fase de classificação e em particular, não afeta a implementação de hardware. Para a solução final será sempre aconselhável utilizar o maior valor para este parâmetro observando, no entanto, que o efeito no algoritmo de treino da SVM é quadrático, de forma que o tempo de treino será o limitante deste parâmetro⁶. Para as análises subsequentes, adotou-se *#training_set* = $100 \times 11 \times 6 = 6600$ como parâmetro *default* por proporcionar tempo de execução razoável e boa taxa de acertos.

⁶Para o presente trabalho o treinamento é feito *offline*.

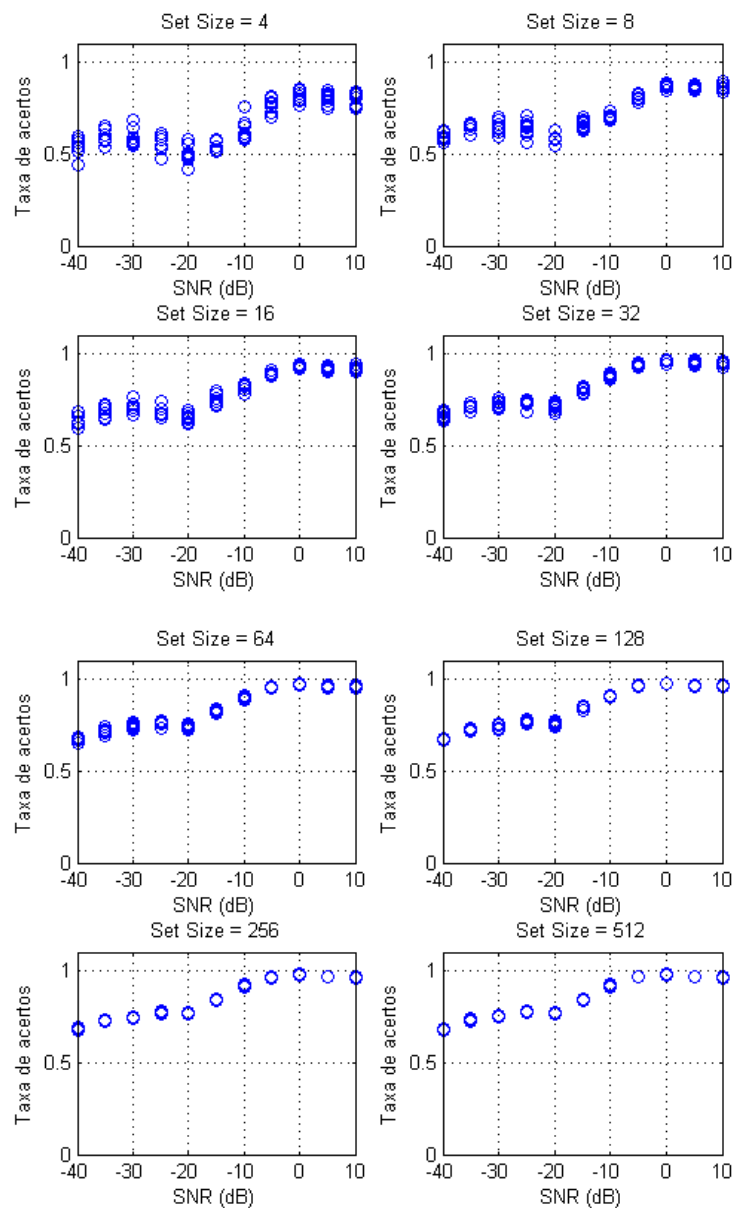


Figura 26 – Taxa de acertos da classificação para diferentes tamanhos do conjunto de treino.

5.3.2 Robustez ao Ruído

Nesta seção consideram-se dois *frameworks* para avaliar o impacto da SNR na taxa de acertos da classificação. No *framework multi-condition* treinou-se um classificador⁷ considerando-se todos os 11 valores de SNR no mesmo conjunto de treino. No *framework matched*, treinaram-se 11 classificadores, cada um com (exclusivamente) um dentre os 11 valores de SNR considerados. Para os demais parâmetros, os valores *default* foram utilizados.

O *framework matched* representa uma situação fictícia onde o classificador seria capaz de identificar o valor da SNR do sinal⁸. O uso de tal *framework* tem dois propósitos: determinar a taxa de acertos máxima que o classificador (baseado em SVM) consegue atingir e comparar tal taxa com aquela que um classificador treinado simultaneamente para mais valores de SVM consegue atingir.

A Figura 27 mostra os resultados, em termos de taxa de acertos, para ambos *frameworks*. Os resultados rotulados como "Matched" correspondem às máximas taxas de acertos obtidas pelo *framework matched*. Como seria de se esperar, para cada valor de SNR, a maior taxa de acertos foi obtida pelo classificador que foi treinado para tal SNR, sendo 100% na maior parte dos casos. Contudo, examinando cada classificador deste *framework*, observa-se que quando treinado exclusivamente com um valor de SNR, o classificador apresenta baixa taxa de acertos para os demais valores de SNR. Para ilustrar tal comportamento, a Figura 27 mostra as taxas de acertos para os classificadores que foram treinados com SNR = -15 dB e com SNR = 10 dB. O classificador treinado com SNR = 10 dB apresenta taxas de acertos de 0,99, 1,0 e 1,0 para SNR = {0, 5, 10}, respectivamente, mas para valores mais baixos de SNR sua taxa de acertos é baixa. O classificador treinado com SNR = -15 dB, por sua vez, apresenta taxas de acertos maiores que 0,79 para SNR = {-10, -5, 0, 5, 10}, mas para SNR = -20 dB a taxa de acertos é 0,25, a qual é próxima da mínima que ele atinge, sido significativamente pior do que o classificador treinado com SNR = 10 dB. Em Fehske, Gaedert e Reed (2005), são mostrados resultados similares que apoiam o *framework multi-condition*.

Uma vez que o *framework matched* constitui uma solução fictícia, uma alternativa pragmática poderia ser treinar um classificador para um único valor de SNR. Entretanto, segundo os resultados apresentados no parágrafo

⁷Para esta seção e para as seções 5.3.3, 5.3.4 e 5.4, considera-se um classificador o conjunto de 6 SVMs (para as 6 modulações consideradas). O resultado de taxa de acertos para cada valor de SNR é obtido através de uma média aritmética do resultado de cada SVM.

⁸Além disso, o *framework matched* faria sentido caso fosse possível treinar o classificador com infinitos valores distintos de SNR.

anterior, tal solução provê uma taxa de acertos baixa. Então, a solução mais viável parece ser treinar o classificador com um número distinto de valores de SNR, ou seja, o *framework multi-condition*.

Na Figura 27 observa-se que a taxa de acertos do *framework multi-condition* é menor para valores baixos de SNR e maior para valores altos de SNR. Particularmente, ele atinge sua máxima taxa de acertos (0,99) quando $\text{SNR} = 0 \text{ dB}$ e não em $\text{SNR} = 10 \text{ dB}$, como seria de se esperar. Além disso, a taxa de acertos para todo o intervalo de valores de SNR é razoável (0,67), sobretudo quando comparada àquela taxa de um dos classificadores do *framework matched*.

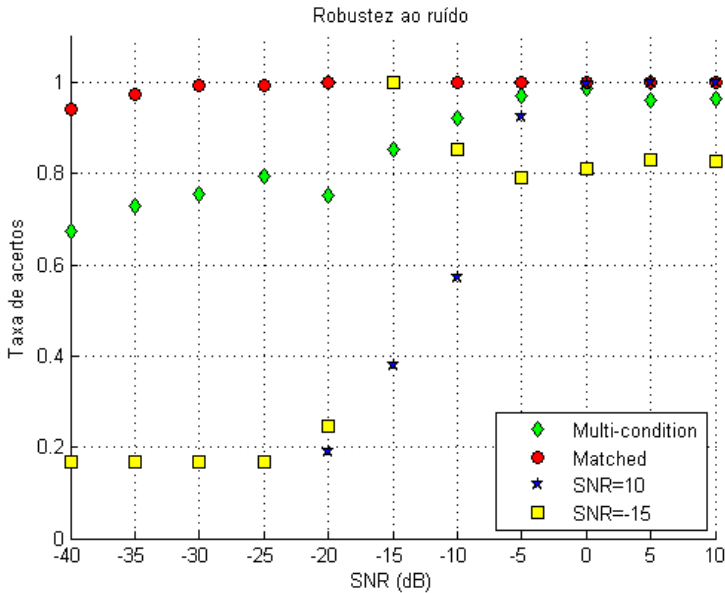


Figura 27 – Taxa de acertos da classificação para *frameworks multi-condition* e *matched*.

5.3.3 Tempo e Taxa de Acertos da Classificação

Nesta subseção, o impacto da quantidade de amostras do sinal (comprimento da sequência de treino/teste) na taxa de acertos da classificação é analisada. Para tanto, são utilizados os parâmetros *default* apresentados na seção 5.3, com exceção do comprimento da sequência (*seq_len*) de treino/teste,

o qual é variado da seguinte forma:

$$seq_len = \{75, 150, 300, 600, 1200\}$$

Como anteriormente, a taxa de acertos da classificação é avaliada em função da SNR, apesar de a sequência de treino considerar todos os valores de SNR de uma só vez (*framework multi-condition*).

Os resultados são mostrados na Figura 28. Observa-se que aumentando o comprimento da sequência, a taxa de acertos da classificação aumenta drasticamente, sobretudo para valores de SNR muito baixos. Para $seq_len = 1200$, a taxa de acertos da classificação para $SNR = -40$ dB é em torno de 95%. Tal comportamento pode ser explicado pelo fato do impacto do ruído Gaussiano reduzir, à medida que mais amostras são adicionadas ao histograma.

Se por um lado o uso de um valor muito grande para seq_len melhora a taxa de acertos, por outro, aumenta o tempo de classificação porque a SVM deve adquirir um maior número de amostras do sinal antes de entregar o resultado da classificação. Portanto, esse parâmetro deve ser escolhido com cuidado e de acordo com as restrições impostas pelo ambiente em que o *Cognitive Radio* (CR) irá operar.

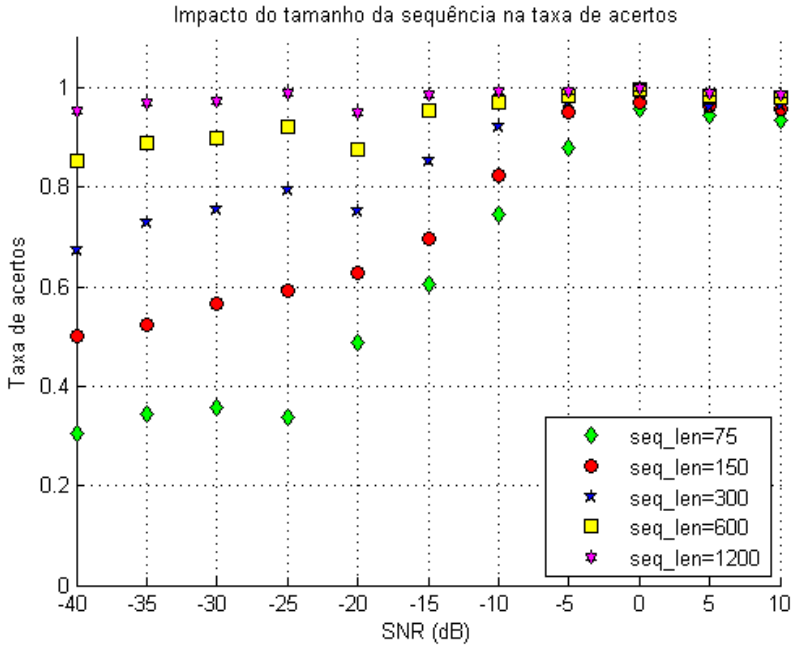


Figura 28 – Taxa de acertos da SVM para diferentes *comprimentos da sequência* em função da *SNR*.

5.3.4 Granularidade do Histograma

Nesta subseção analisa-se o impacto do número de *bins* na taxa de acertos da SVM. Para tanto, a SVM foi treinada e testada usando-se os parâmetros *default* apresentados na seção 5.3 exceto o número de *bins*, o qual foi variado dentro do seguinte conjunto:

$$nb_bins = \{4^2, 8^2, 16^2, 32^2\}$$

Os resultados da taxa de acertos da classificação são mostrados na Figura 29. Nota-se que o número de *bins* do histograma impacta drasticamente na taxa de acertos da SVM. O número de *bins* no histograma corresponde à dimensão do espaço no qual a SVM realiza a classificação. Desta forma, aumentar a dimensão dá mais "liberdade" à máquina e a capacidade de aprendizado é aumentada (VAPNIK, 1999).

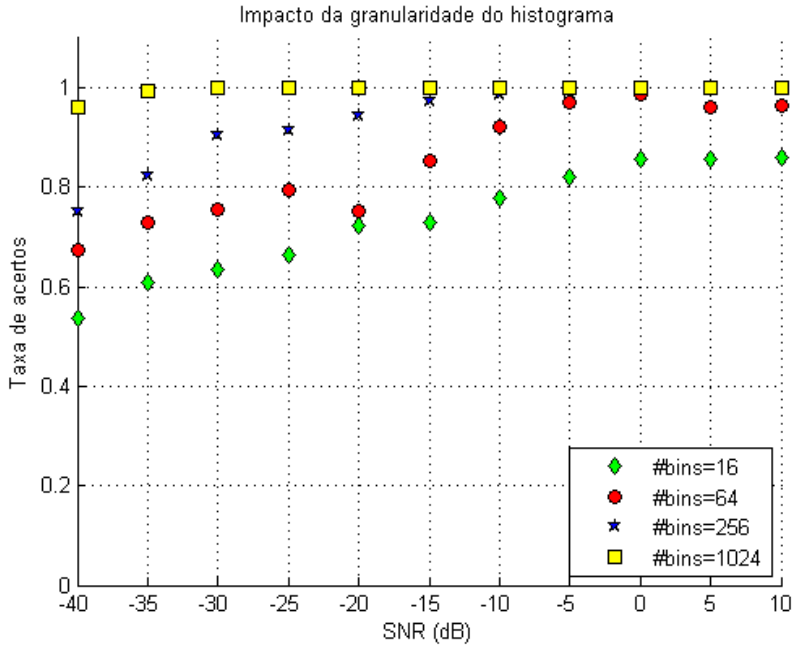


Figura 29 – Taxa de acertos da SVM para diferentes valores de *nb_bins*.

Como visto na subseção 2.3.3, o número de *bins* corresponde ao número de coeficientes utilizados na fase de teste (classificação). Assim, há um compromisso entre a taxa de acertos e a quantidade de recursos de hardware necessários, o qual deve ser levado em conta no momento de se projetar uma solução. Por exemplo, usando-se *nb_bins*= 64, como adotado nos parâmetros *default*, o classificador atinge uma taxa de acertos de apenas 0,67 para $\text{SNR} = -40$ dB. Por outro lado, quando *nb_bins*= 1024, o classificador atinge taxa de acertos de 0,96 para $\text{SNR} = -40$ dB, a um custo de 16 vezes mais memória (para armazenar os coeficientes).

5.4 COMPARAÇÃO DA INFLUÊNCIA DA GRANULARIDADE DE HISTOGRAMAS 1-D VERSUS HISTOGRAMAS 2-D NA TAXA DE ACERTOS DA CLASSIFICAÇÃO

Esta seção apresenta uma avaliação da influência do número de *bins* na taxa de acertos de classificadores que utilizam extratores de parâmetros

baseados em histogramas 1-D e a consequente comparação com as análises da seção 5.3.4 (referentes a histogramas 2-D). Para tanto, são usados os parâmetros *default* apresentados na seção 5.3, exceto pelo número de *bins*, o qual é variado seguindo o conjunto:

$$nb_bins = \{2 \times 8, 2 \times 32, 2 \times 128, 2 \times 512\}$$

Os resultados de classificação para HISTO-1D são exibidos na Figura 30.

Comparando-se a Figura 29 com a Figura 30 é possível notar que o classificador baseado em HISTO-1D apresenta resultados piores do que os classificadores HISTO-2D, sobretudo em valores de SNR baixos e com menos de 1024 *bins*. A Tabela 5, que apresenta a taxa de acertos global (isto é, uma média de todos os resultados de todos os valores de SNR considerando um determinado valor de *bin*), mostra que o resultados dos classificadores HISTO-1D são piores do que os resultados dos classificadores HISTO-2D chegando a apresentar uma diferença de 9,5% para 256 *bins*. Em geral, o histograma 2-D tem um desempenho melhor do que o histograma 1-D para o mesmo número de coeficientes. Isto demonstra que a correlação da informação discutida na subseção 5.1 é mais importante do que a maior resolução fornecida pelo histograma 1-D (veja Figuras 18, 19, 20 e 21).

Essa constatação reforça o uso de HISTO-2D, sobretudo se levarmos em consideração que para CR é muito importante detectarmos sinais de usuários primários de baixa intensidade nos quais a SNR geralmente é baixa.

Na próxima seção as arquiteturas de hardware para classificadores SVM-HISTO-2D são analisadas e comparadas com as arquiteturas para SVM-HISTO-1D. As modificações necessárias na arquitetura da SVM-HISTO-1D (apresentada na seção 4.2) para implementar SVM-HISTO-2D são discutidas.

	1-D	2-D
16 <i>bins</i>	0,670	0,734
64 <i>bins</i>	0,775	0,851
256 <i>bins</i>	0,841	0,936
1024 <i>bins</i>	0,993	0,996

Tabela 5 – Taxa de acertos global para histogramas 1D e histogramas 2D

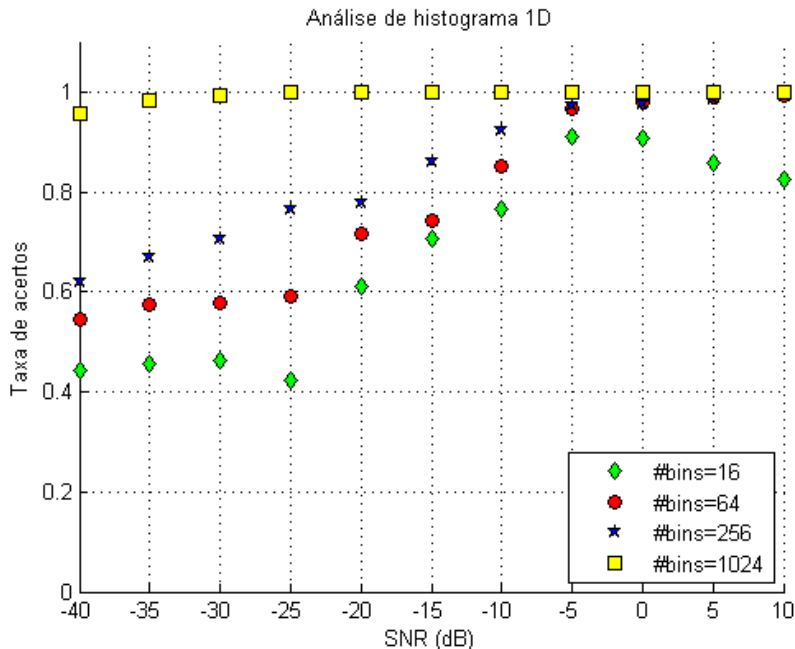


Figura 30 – Taxa de acertos para histogramas 1-D para diferentes valores de nb_bins .

5.5 ARQUITETURAS PARA CLASSIFICADORES SVM E HISTOGRAMAS 2-D

Como apresentado na seção 5.1, um histograma 2-D é obtido correlacionando os valores da parte real e imaginária da amostra, de forma que o valor resultante é um ponto em uma "matriz" (valor do coeficiente) que representa a amostra presente na entrada do classificador. Desta forma, para uma arquitetura baseada em histograma 2-D, apenas uma memória de coeficientes é necessária, o que torna a implementação de hardware para classificadores HISTO-2D ligeiramente mais eficiente do que aquela utilizada para classificadores HISTO-1D. Para cada amostra, apenas um acesso à memória é necessário e apenas uma adição é realizada.

Da mesma forma que para os classificadores HISTO-1D, é possível construir um *datapath* sem multiplicadores, correlacionando a amostra de entrada com o valor de coeficiente armazenado na memória de coeficientes (ver

subseção 2.3.3). A Figura 31 apresenta o *datapath* genérico do classificador SVM-HISTO-2D. Esta figura também mostra que é necessário apenas um somador e um registrador (comparar com a Figura 11), ambos de 32 bits. Da mesma forma que para a SVM-HISTO-1D, o valor b é fornecido via somador logo no início da classificação, a cada novo ciclo de classificação. Os m bits mais significativos das partes real e imaginária do sinal são concatenados para formar o endereço do coeficiente na memória de coeficientes. A saída da memória apresentará o valor do coeficiente a ser acumulado no registrador de saída **ACC**. O processo é repetido até que o número total de amostras para classificação seja recebido.

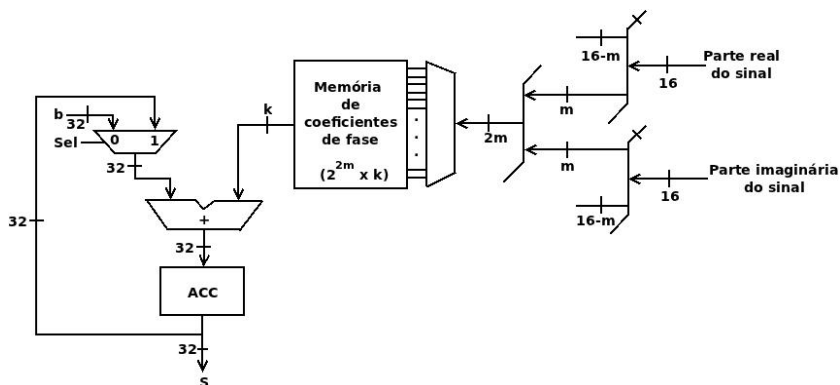


Figura 31 – Diagrama em blocos nível RTL de um bloco SVM.

Na Figura 31 o valor m está diretamente relacionado com a quantidade de coeficientes obtidos da fase de treino, este último sendo igual ao número de *bins* do conjunto de treino. O valor k é o número de bits utilizados para armazenar os mesmos coeficientes (nos experimentos de síntese para FPGA apresentados a seguir, esse valor foi variado no conjunto {8, 12, 16}).

É importante notar que o tamanho total da memória permanece o mesmo tanto para uma SVM para HISTO-1D quanto para uma SVM para HISTO-2D, dado um mesmo número de *bins*, com a diferença que para HISTO-1D são necessários dois blocos de memória.

A Figura 32 mostra o diagrama em blocos da arquitetura de um classificador SVM-HISTO-2D para as 6 modulações testadas nessa etapa do trabalho. Particularmente, observa-se a utilização do esquema *One Against All* (OAA) melhorado, apresentado na seção 4.6.

O classificador resultante foi descrito em linguagem Verilog e sintetizado para um FPGA da família Cyclone II da altera (*EP2C35F672C6*) utili-

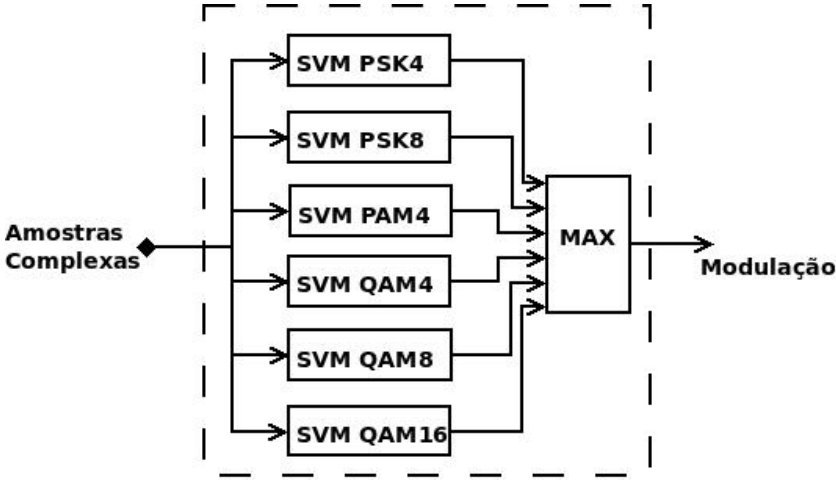


Figura 32 – Diagrama de blocos de um classificador **SVM-HISTO-2D** para as 6 modulações digitais consideradas neste capítulo.

zando a ferramenta Quartus II. Variações do número de *bins* e do número de bits foram realizadas com o objetivo de avaliar a influência no desempenho (frequência máxima de funcionamento) e no total de recursos utilizados no dispositivo selecionado, conforme apresentado na próxima seção (em conjunto com a comparação com o classificador SVM-HISTO-1D). A Tabela 6 mostra o resultado de síntese no que se refere ao uso de bits de memória, quando $k = 8\ bits^9$.

Número de Bins	Bits de memória
16	768(<1%)
64	3.072(<1%)
256	12.288(3%)
1024	49.152(10%)

Tabela 6 – Utilização de bits de memória do FPGA.

Como é possível verificar, mesmo para o pior caso (1024 *bins*), a quantidade de recursos utilizados não passa de 10% do total de blocos de memória disponível no dispositivo selecionado. A próxima seção apresenta o resultado

⁹A memória é o bloco que ocupa maior quantidade de recursos de hardware.

completo de síntese para FPGA para a SVM-HISTO-2D, comparando-o com o resultado de síntese para SVM-HISTO-1D.

5.5.1 Comparação dos Resultados de Síntese para HISTO-1D e HISTO-2D

Para uma melhor avaliação, além do número de coeficientes, que influencia o número de posições de memória (2^{2m} na Figura 31), também foi variado o tamanho da palavra de memória (k na Figura 31). Com isso, é possível ter-se uma visualização mais clara do total de recursos necessários para cada abordagem.

As Tabelas 7 e 8 mostram os resultados de síntese para o FPGA selecionado, tanto para a SVM-HISTO-1D quanto para a SVM-HISTO-2D.

		16 bins	64 bins	256 bins	1024 bins
8 bits	SVM-HISTO-1D	1064	1062	1063	1063
	SVM-HISTO-2D	1008	1008	1010	1008
12 bits	SVM-HISTO-1D	1087	1087	1089	1086
	SVM-HISTO-2D	1010	1009	1009	1012
16 bits	SVM-HISTO-1D	1114	1111	1115	1112
	SVM-HISTO-2D	1010	1010	1009	1012

Tabela 7 – Utilização de *logical elements* nos classificadores **SVM-HISTO-1D** e **SVM-HISTO-2D**.

A Tabela 7 apresenta o total de elementos lógicos resultantes da síntese dos classificadores SVM-HISTO-1D e SVM-HISTO-2D, quando o número de *bins* e de bits é variado. Observa-se que para todas as combinações de número de *bins* e número de bits, o classificador SVM-HISTO-1D requer mais elementos lógicos do que o classificador SVM-HISTO-2D. Entretanto, tal diferença é pequena, entre 4,9% e 9,3%. Especula-se que tal diferença deva-se ao fato do classificador SVM-HISTO-1D necessitar de mais hardware pra controle da memória, uma vez que utiliza dois blocos de memória, ao invés de somente um bloco. Por outro lado, dado um tipo de classificador, projetado considerando-se um certo número de bits para os coeficientes, nota-se que o número de elementos lógicos resultantes da síntese é praticamente constante, e portanto, independente do número de *bins*. Na realidade, esta

variação é simplesmente efeito dos algoritmos de síntese da ferramenta, uma vez que o comprimento do registrador **ACC** e do somador foi fixado quando do projeto das arquiteturas. Semelhante consideração pode ser feita quando se analisa os resultados da síntese para um tipo de classificador e um certo número de *bins*.

A Tabela 8 apresenta resultados de utilização de bits de memória iguais para arquiteturas SVM-HISTO-1D e para SVM-HISTO-2D. Para fazer uma comparação mais justa entre as arquiteturas, foi utilizada uma opção de compilação do Quartus II que impede a otimização de recursos de memória quando a quantidade de bits é pequena¹⁰. Desta forma, foi forçada a situação em que todos os bits de memória são alocados em elementos de memória, independente do total de memória necessário.

Esse resultado de utilização da memória era esperado, pois a mesma quantidade de elementos de memória deve ser alocada quando o número de *bins* é o mesmo, independentemente do tipo de classificador (SVM-HISTO-1D e SVM-HISTO-2D).

	16 <i>bins</i>	64 <i>bins</i>	256 <i>bins</i>	1024 <i>bins</i>
8 bits	768	3072	12288	49152
12 bits	1152	4608	18432	73728
16 bits	1536	6144	24576	98304

Tabela 8 – Utilização de bits de memória para HISTO-1D e HISTO-2D.

5.6 CONCLUSÕES DO CAPÍTULO

De uma forma geral, os resultados apresentados neste capítulo permitiram uma melhor avaliação da utilização de histogramas 2-D para a tarefa de AMC utilizando SVM e histogramas. Mais importante de tudo, a avaliação realizada permitiu concluir que para ambientes não cooperativos (*blind estimation*), comuns em sistemas de CR, a abordagem utilizando histogramas 2-D permite resultados melhores em condições de baixos valores de SNR devido ao fato de capturar a correlação entre as partes real e imaginária do sinal. As investigações de tamanho de conjunto de treino permitiram estabelecer que, a partir de 256 histogramas não há melhora significativa na taxa de acertos da classificação. Da mesma forma, comprovou-se que a utilização

¹⁰Esta opção de compilação (*Allow any RAM size for recognition = Off* em *Assignments* → *Settings* → *Analysis & Synthesis* → *More Settings*) impede que o Quartus II aloque registradores de 1 bit (*flip-flops*) para funcionarem como bits de memória quando a quantidade total de memória é pequena.

do *framework multi-condition* apresenta melhores resultados, conforme já declarado por (FEHSKE; GAEDDERT; REED, 2005). Como esperado, a seção 5.3.3 mostrou que quanto maior a quantidade de amostras do sinal, melhor será a taxa de acertos, da mesma forma como ocorre com a quantidade de *bins* do histograma: quanto mais *bins*, melhor a classificação.

Outra conclusão importante que as análises deste capítulo possibilitaram foi que a assertiva de não sincronismo entre transmissor e receptor resultou em resultados de classificação mais precisos. Para efeitos ilustrativos, compare o valor de $SNR = 0$ dB para a curva para OAA melhorado, 16 *bins* da Figura 17 e a curva com a mesma quantidade de *bins* da Figura 30.

O próximo capítulo apresenta as conclusões gerais deste trabalho de mestrado e elenca possibilidades de futuros trabalhos na área.

6 CONCLUSÕES

Atualmente, o rádio cognitivo (*Cognitive Radio* (CR)) mostra-se como a solução mais pragmática para contornar o problema de escassez do espectro de rádio frequências. Entretanto, o sucesso do CR passa pela implementação de técnicas eficientes de monitoramento espectral (*Spectrum Sensing* (SS)), dentre as quais, a Classificação Automática de Modulações (*Automatic Modulation Classification* (AMC)) desponta como uma das mais interessantes, sendo objeto de muitas pesquisas contemporâneas. Sendo a AMC uma tarefa computacionalmente intensiva e que precisa ser executada muitas vezes, o desenvolvimento de hardware dedicado surge como uma necessidade para viabilizar a adoção do CR.

Dentre as técnicas possíveis de serem aplicadas para a tarefa de AMC, a técnica de classificação baseada em características do sinal tem se mostrado a mais promissora para implementação em hardware. A modelagem básica desta técnica prevê uma etapa de extração de características e uma etapa de classificação propriamente dita. Muitas características específicas dos sinais de rádio frequências podem ser utilizadas para identificar um determinado tipo de modulação. Contudo, o custo computacional necessário para extrair tais características é um fator fundamental para o projeto de AMC em hardware dedicado. Outra etapa que agrega custo computacional à tarefa de AMC é a classificação em si. Portanto, otimizar a solução conjunta das etapas de extração de características e de classificação leva a uma solução de hardware dedicado mais eficiente.

Inspirando-se no trabalho de Cardoso, Castro e Klautau (2013), adotou-se o histograma do sinal como característica a ser utilizada para a classificação e a máquina de aprendizado *Support Vector Machine* (SVM) como classificador. Tal abordagem permitiu descrever um hardware bastante simples, o qual foi apresentado no capítulo 4. No mesmo capítulo foram apresentadas variações de esquemas para solucionar o problema de classificação de múltiplas classes utilizando um classificador binário (como a SVM). O esquema ***One Against All* (OAA) melhorado** apresentou resultados de taxa de acertos de classificação até 26% melhores (para 0 dB) do que o esquema OAA, com consumo de energia apenas 2% maior do que este último. Os experimentos realizados no capítulo 4 permitiram definir uma arquitetura básica de hardware dedicado, a qual foi utilizada no restante do trabalho.

Em ambiente de CR é comum encontramos a situação em que o transmissor e o receptor não cooperam entre si, isto é, o receptor não conhece a taxa de transmissão na qual o transmissor opera. Entretanto, a maioria dos trabalhos em AMC assume que transmissor e receptor operam com a mesma

taxa. Os experimentos apresentados no capítulo 5 assumem que a transmissão é não-cooperativa, o que representa um importante diferencial para a pesquisa em AMC. Como sub-produto de tal assertiva, foi possível constatar quão robusto é o método de classificação adotado quando considera-se um ambiente mais realista como o não-cooperativo.

No capítulo 5 também constatou-se que capturar a relação entre as partes real e imaginária da amostra complexa do sinal leva a um resultado de classificação que é melhor do que aquele obtido quando se aumenta o número de *bins*, o qual, em última instância, significa aumentar o custo do hardware.

A captura da correlação entre as partes real e imaginária do sinal foi possível graças à adoção de histogramas 2-D, ao invés de histogramas 1-D (como utilizado no trabalho de Cardoso, Castro e Klautau (2013)), o que também consiste em uma contribuição do presente trabalho. A comprovação da importância da correlação entre as partes real e imaginária do sinal foi realizada através de experimentos que avaliaram a influência na taxa de acertos da classificação de parâmetros como tamanho do conjunto de treino, relação sinal-ruído (*Signal-to-Noise Ratio* (SNR)), comprimento da sequência e número de *bins*.

Nos experimentos com o tamanho do conjunto de treino determinou-se o valor de 256 amostras como sendo um valor ótimo para esse parâmetro. Observou-se que abaixo deste valor os resultados de classificação não são estáveis (fato esse que pode ser facilmente identificado na Figura 26). Por outro lado, aumentar esse valor significa aumentar o tempo de treino de forma quadrática, o que não tem maior impacto para o presente trabalho, uma vez que o treinamento é feito *off-line*. Porém, em uma possível implementação na qual se queira fazer o treinamento *on-line* (no hardware dedicado), o tempo de treinamento passa a ser importante. Uma situação onde se faz necessário fazer o treinamento *on-line* da SVM no contexto de CR seria o de reconfigurar o hardware do receptor para novas formas de onda. A reconfigurabilidade do hardware é um dos principais aspectos que diferenciam CR de *Software-Defined Radio* (SDR) e se apresenta com outra frente de pesquisas para a viabilização do CR.

Em relação à análise de robustez ao ruído (SNR), os experimentos mostraram que o classificador apresenta resultados de classificação muito melhores se este for treinado para toda a faixa de valores de SNR na qual irá operar (*framework multi condition*).

Já a análise do comprimento da sequência mostrou que ao aumentar o número de amostras utilizadas para compor o histograma, a taxa de acertos da classificação apresenta uma significativa melhora. Por outro lado, aumentar o comprimento da sequência significa que o classificador deverá esperar até que todas as amostras cheguem, antes de realizar a classificação, o que significa

atrasar a decisão de sair de um canal ocupado por um usuário primário. (O tempo para sair de um canal que estava livre e passou a ser utilizado por um usuário primário é um dos principais requisitos do CR).

Semelhantemente ao aumento do comprimento da sequência, aumentar a quantidade de *bins* do histograma também tem forte influência na taxa de acertos da classificação: quanto maior o número de *bins* do histograma, maior é a taxa de acertos da classificação. A contrapartida é que aumentar o número de *bins* aumenta também o número de coeficientes a serem armazenados e isso, por sua vez, tem impacto no hardware resultante.

Com relação ao comprimento da sequência de teste e à quantidade de *bins* do histograma, uma melhora significativa ocorre quando esses parâmetros são aumentados. Porém, ambos tem impacto negativo no tempo de classificação e no tamanho do hardware resultante.

Mais importante de tudo, a investigação considerando que o receptor não está sincronizado com o transmissor surpreendentemente mostrou que essa técnica apresenta resultados de classificação melhores do que aqueles obtidos quando se considera transmissor e receptor sincronizados. Esse fato é facilmente observado ao se comparar os resultados de classificação apresentados na seção 4.6 com os resultados apresentados na seção 5.4 para o valor de SNR = 0 dB. Mesmo considerando 16 *bins*, percebe-se que o segundo caso apresenta resultados de classificação muito melhores que o primeiro.

Os experimentos realizados para a definição da arquitetura de hardware dedicado para classificadores baseados em histogramas e SVM (capítulo 4) e o aprofundamento de conhecimentos referentes à aplicação de SVM em AMC levaram à conclusão de que o *kernel* linear apresenta resultados satisfatórios a um baixo custo de hardware. Por esse motivo, o objetivo 2 (investigar a adoção de SVMs com *kernel* não linear), proposto na seção 1.6 desta dissertação, não foi cumprido. Ao invés de investigar SVMs com *kernel* não linear, cujo custo final de hardware muito provavelmente seria maior, decidiu-se investigar o uso da técnica combinada de histogramas com SVMs para AMC em ambientes não cooperativos, apresentada no capítulo 5.

6.1 TRABALHOS FUTUROS

Durante o desenvolvimento desta pesquisa percebeu-se que outros tópicos poderiam ser investigados utilizando-se a infraestrutura experimental desenvolvida:

- Análise do impacto do ruído de quantização das amostras. Para sistemas de comunicação, o ruído de quantização quase sempre deve ser levado em conta. Para o presente trabalho de mestrado, as amostras do

sinal foram representadas por um número inteiro de 16 bits, seguindo a tendência das demais implementações em hardware dedicado na área. Contudo, uma investigação do ruído de quantização se faz necessária, como uma complementação do presente trabalho.

- Análise do impacto do canal de transmissão. A literatura atual utiliza, em sua grande maioria, o modelo de canal Gaussiano, que é um modelo simplificado de representação do canal de transmissão. No entanto, sabe-se que para uma representação mais fiel do canal de transmissão outras influências devem ser consideradas, como por exemplo, influências de múltiplos caminhos, desvanecimento do canal, etc. Para tanto, um modelo de canal mais aprimorado deve ser utilizado, o que elevaria o nível de dificuldade das análises.
- Análise de desempenho em modelo físico. As análises apresentadas neste trabalho de mestrado utilizaram ferramentas matemáticas como o software Matlab[®]. Um possível desdobramento seria realizar experimentos práticos com alguma plataforma real de SDR como por exemplo, a *Universal Software Radio Peripheral* (USRP) (USRP, 2014).

REFERÊNCIAS

- ANJUM, O. et al. State of the art baseband dsp platforms for software defined radio: A survey. *EURASIP Journal on Wireless Communications and Networking*, Springer, v. 2011, n. 1, p. 1–19, 2011.
- ARSLAN, H. *Cognitive radio, software defined radio, and adaptive wireless systems*. [S.l.]: Springer, 2007.
- BACHE, K.; LICHMAN, M. *UCI Machine Learning Repository*. 2013. <<http://archive.ics.uci.edu/ml>>. Acessado em 15/09/14.
- BURGES, C. J. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, Springer, v. 2, n. 2, p. 121–167, 1998.
- BYVATOV, E. et al. Comparison of support vector machine and artificial neural network systems for drug/nondrug classification. *Journal of Chemical Information and Computer Sciences*, ACS Publications, v. 43, n. 6, p. 1882–1889, 2003.
- CARDOSO, C.; CASTRO, A. R.; KLAUTAU, A. An efficient FPGA IP core for automatic modulation classification. *Embedded Systems Letters*, IEEE, v. 5, n. 3, p. 42–45, 2013. ISSN 1943-0663.
- CHANG, C.-C.; LIN, C.-J. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, ACM, v. 2, n. 3, p. 27, 2011.
- CORMEN, T. H. et al. *Introduction to algorithms*. The MIT Press, 2009.
- CORTES, C.; VAPNIK, V. Support-vector networks. *Machine learning*, Springer, v. 20, n. 3, p. 273–297, 1995.
- DELAHAYE, J.-P. et al. Software radio and dynamic reconfiguration on a dsp/fpga platform. *Frequenz*, v. 58, n. 5-6, p. 152–159, 2004.
- DICK, C. H.; PEDERSEN, H. M. Design and implementation of high-performance fpga signal processing datapaths for software defined radios. In: *Embedded Systems Conference*. [S.l.: s.n.], 2001. p. 1–16.
- DOBRE, O. A. et al. Survey of automatic modulation classification techniques: classical approaches and new trends. *Communications, IET*, IET, v. 1, n. 2, p. 137–156, 2007.

Ettus Research. *USRP family of products*. 2014.

<<http://www.ettus.com/>>. Acessado em 10/20/14.

FCC, E. Docket no. 03-108. *Facilitating opportunities for flexible, efficient, and reliable spectrum use employing cognitive radio technologies, FCC Report and Order adopted (March 10, 2005)*, 2005.

FEHSKE, A.; GAEDDERT, J.; REED, J. A new approach to signal classification using spectral correlation and neural networks. In: IEEE. *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*. [S.l.], 2005. p. 144–150.

HALL, M. et al. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, ACM, v. 11, n. 1, p. 10–18, 2009.

HANSON, R.; STUTZ, J.; CHEESEMAN, P. *Bayesian classification theory*. [S.l.]: NASA Ames Research Center, Artificial Intelligence Research Branch, 1991.

HARADA, H. A software defined cognitive radio prototype. In: IEEE. *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*. [S.l.], 2007. p. 1–5.

HARUYAMA, S. FPGA in the software radio. *IEEE communications Magazine*, v. 37, n. 2, p. 108–112, Feb 1999. ISSN 0163-6804.

HAYKIN, S. *Communication systems*. [S.l.]: John Wiley & Sons, 2008.

HEARST, M. A. et al. Support vector machines. *Intelligent Systems and their Applications, IEEE*, IEEE, v. 13, n. 4, p. 18–28, 1998.

HSU, C.-W.; LIN, C.-J. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, IEEE, v. 13, n. 2, p. 415–425, 2002.

HU, H.; SONG, J.; WANG, Y. Signal classification based on spectral correlation analysis and SVM in cognitive radio. In: IEEE. *Advanced Information Networking and Applications, 2008. AINA 2008. 22nd International Conference on*. [S.l.], 2008. p. 883–887.

IMMONEN, A. et al. A reconfigurable multi-standard radio platform. In: SPRINGER. *Mobile Multimedia Communications*. [S.l.], 2012. p. 661–671.

KLAUTAU, A. *Digital Communications and Signal Processing*. [s.n.], ca. 2014. <<http://www.aldebaro.ufpa.br/#home>>. Acessado em 11/08/14.

KOLODZY, P. Spectrum policy task force report. *Federal Communications Commission, Washington, DC, Rep. ET Docket*, n. 02-135, 2002.

LORENA, A. C.; CARVALHO, A. C. D. Evolutionary tuning of SVM parameter values in multiclass problems. *Neurocomputing*, Elsevier, v. 71, n. 16, p. 3326–3334, 2008.

LUXBURG, U. V.; SCHÖLKOPF, B. Statistical learning theory: Models, concepts, and results. *arXiv:0810.4752*, Cornell University Library, 2008.

Mentor Graphics. *ModelSim - Leading Simulation and Debugging - Mentor Graphics*. 2011. <<http://www.mentor.com/products/fpga/model>>. Acessado em 09/02/14.

MICHIE, D.; SPIEGELHALTER, D. J.; TAYLOR, C. C. Machine learning, neural and statistical classification. Ellis Horwood, 1994.

MINDEN, G. J. et al. Kuar: A flexible software-defined radio development platform. In: IEEE. *New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on*. [S.l.], 2007. p. 428–439.

MULLER, F. C.; CARDOSO, C.; KLAUTAU, A. A front end for discriminative learning in automatic modulation classification. *Communications Letters, IEEE*, IEEE, v. 15, n. 4, p. 443–445, 2011.

Nokia. *Cognitive Radio Liberating Spectrum to Enable New Opportunities*. 2014. <http://research.nokia.com/cognitive_radio>. Acessado em 10/20/14.

OUEDRAOGO, G. S.; GAUTIER, M.; SENTIEYS, O. Frame-based modeling for automatic synthesis of fpga-software defined radio. In: IEEE. *Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM), 2014 9th International Conference on*. [S.l.], 2014. p. 341–346.

PROAKIS, J. G. Book. *Digital communications*. [S.l.]: McGraw-Hill New York, 2008. ISBN 9780072957167 0072957166.

RAMACHER, U. et al. Software-defined radio prospects for multistandard mobile phones. *IEEE Computer*, v. 40, n. 10, p. 62–69, 2007.

RIYANTO, B. et al. Software architecture of software-defined radio (SDR). In: . [S.l.: s.n.].

ROY, F. L. et al. A survey of frameworks & open environments applied to cognitive radio design. *IJETAE*, v. 4, n. 3, 2014.

SKLAR, B. *Digital communications*. [S.l.]: Prentice Hall NJ, 2001.

SMOLA, A. J.; SCHÖLKOPF, B. A tutorial on support vector regression. *Statistics and computing*, Springer, v. 14, n. 3, p. 199–222, 2004.

SORATO, E. et al. Real-time digital modulation classification based on support vector machines. In: *2013 26th Symposium on Integrated Circuits and Systems Design (SBCCI)*. [S.l.]: IEEE, 2013. p. 1–6. ISBN 978-1-4799-1132-5.

SORATO, E. et al. VLSI architectures for digital modulation classification using support vector machines. In: IEEE. *Circuits and Systems (LASCAS), 2013 IEEE Fourth Latin American Symposium on*. [S.l.], 2013. p. 1–4.

SU, W.; XU, J. L.; ZHOU, M. Real-time modulation classification based on maximum likelihood. *Communications Letters, IEEE*, IEEE, v. 12, n. 11, p. 801–803, 2008.

SYNOPSYS. *Synopsys Design Compiler*. 2011. <<http://www.synopsys.com/Tools/Implementation/RTLSynthesis/DesignCompiler/Pages/default.aspx>>. Acessado em 28/05/14.

TABATABAEI, T. S.; KRISHNAN, S.; ANPALAGAN, A. SVM-based classification of digital modulation signals. In: IEEE. *Systems man and cybernetics (SMC), 2010 IEEE international conference on*. [S.l.], 2010. p. 277–280.

USRP. *Universal Software Radio Peripheral (USRP)*. 2014. <<http://www.radio-assa.org.au/book/export/html/30>>. Acessado em 11/08/14.

VAPNIK, V. N. An overview of statistical learning theory. *Neural Networks, IEEE Transactions on*, IEEE, v. 10, n. 5, p. 988–999, 1999.

WU, Z. et al. Automatic digital modulation recognition based on support vector machines. In: IEEE. *Neural Networks and Brain, 2005. ICNN&B'05. International Conference on*. [S.l.], 2005. v. 2, p. 1025–1028.

XG Technology. *Cognitive Radio Networks*. 2014. <<http://www.xgtechnology.com/technology/cognitive-radio-networks/>>. Acessado em 10/20/14.

XIAO, H. et al. Automatic classification of analog modulation schemes. In: *Radio and Wireless Symposium (RWS), 2012 IEEE*. [S.l.]: IEEE, 2012. p. 5–8. ISSN 2164-2958.

YUCEK, T.; ARSLAN, H. A survey of spectrum sensing algorithms for cognitive radio applications. *Communications Surveys & Tutorials, IEEE*, IEEE, v. 11, n. 1, p. 116–130, 2009.